UNIVERSITAT POLITÈCNICA DE CATALUNYA

DOCTORAL THESIS

Real-time Nonlinear Model Predictive Control for Thermal Management in a Plug-In Hybrid Electric Vehicle

Author:
Jorge LOPEZ SANZ

Supervisors:
Dr. Jesus Andres ALVAREZ FLOREZ
Dr. Juan Manuel MORENO EGUILAZ
Dr. Rafael RUIZ MANSILLA

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Heat Engines Department
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

May 19, 2016

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Assessment results for the doctoral thesis

Academic year:

Full name

Doctoral programme

Structural unit in charge of the programme

Decision of the committee

In a meeting with the examination committee convened for this purpose, the doctoral candidate presented the topic of his/her doctoral thesis entitled

_____

_____.

Once the candidate had defended the thesis and answered the questions put to him/her, the examiners decided to award a mark of:

☐ UNSATISFACTORY ☐ SATISFACTORY ☐ GOOD ☐ VERY GOOD

(Full name and signature) Chairperson		(Full name and signature) Secretary
(Full name and signature) Member	(Full name and signature) Member	(Full name and signature) Member

_____, _____

The votes of the members of the examination committee were counted by the Doctoral School at the behest of the Doctoral Studies Committee of the UPC, and the result is to award the CUM LAUDE DISTINCTION:

☐ YES ☐ NO

(Full name and signature) Chair of the Standing Committee of the Doctoral School	(Full name and signature) Secretary of the Standing Committee of the Doctoral School

Barcelona, _____

“Sometimes a scream is better than a thesis.”

— Manfred Eigen. German biophysical chemist, 1967 Nobel Prize winner.

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona
Heat Engines Department

Doctor of Philosophy

Real-time Nonlinear Model Predictive Control for Thermal Management in a Plug-In Hybrid Electric Vehicle

by Jorge LOPEZ SANZ

Several socioeconomic factors are leading governments to encourage electric powered vehicles. Currently, the bottleneck for electric vehicles mass production lies in the high voltage battery technology. One of the main challenges to ensure batteries safety, comfort, performance and durability requirements is thermal management, since operating at temperatures outside the range specified by the manufacturer, they age prematurely, lead to dangerous and uncontrolled exothermic reactions and/or be incapable of delivering the electric energy demand to move the vehicle. The tendency in the solutions design for thermal management is to use cooling circuits with more and more sophisticated architectures governed by an increasing number of electrical actuators like pumps, fans and solenoid valves. The control of these systems is complex due to their nonlinear behavior, the high number of inputs and outputs and the need of accomplishing multiple goals, usually contradictory, at the same time. In front of this class of problems, conventional control methods are taken to their limit and new optimization based methods, like model predictive control, capable of exploiting the full potential in this kind of systems, are attracting the attention of the sector.

The present thesis deals with the design of a predictive control for the battery and power electronics cooling circuit in a Plug-In Hybrid Electric Vehicle. The main merit of the proposed solution is that the method validation takes places in a prototype on real-time, which, as it will be seen in the state of the art, is one of the usual lacks in most model predictive control publications in the automotive sector. For reaching this settlement, the development of a suitable model of the system and optimization problem definition together with the use of an efficient and robust numerical tool, have been essential and therefore will be addressed exhaustively in this document. Additionally, the validation by means of simulation as well as the design of repeatable driving conditions for comparing the proposed control with the original one in the vehicle will be shown before reaching the final validation and discussion.

Resumen (spanish)

Diversos factores socioeconómicos están llevando a los gobiernos a fomentar los vehículos propulsados eléctricamente. Actualmente, el cuello de botella para la producción en masa del vehículo eléctrico reside en la tecnología de la batería de alta tensión. Uno de los retos principales para asegurar las prestaciones de seguridad, confort, funcionamiento y durabilidad de la batería es la gestión térmica, ya que a temperaturas alejadas de las especificadas por el fabricante, ésta envejece de forma prematura, dar lugar a una peligrosa y descontrolada reacción exotérmica y/o ser incapaz de entregar la energía eléctrica necesaria para mover el vehículo. La tendencia en el diseño de soluciones para la gestión térmica, es la de usar circuitos de refrigeración con arquitecturas cada vez más sofisticadas que implican la necesidad de un mayor número de actuadores eléctricos como bombas, ventiladores y electroválvulas. El control de estos sistemas es complejo debido a su comportamiento no lineal, al elevado número de entradas y salidas y a la necesidad de lograr varios objetivos a la vez a menudo contradictorios. Ante esta clase de problemas, los métodos convencionales de control son llevados a su límite y nuevos métodos basados en optimización, como el control predictivo, capaces de explotar el potencial de este tipo de sistemas, empiezan a atraer la atención del sector.

Esta tesis trata del diseño de un control predictivo para la gestión térmica del circuito de refrigeración de la batería y la electrónica de potencia en un vehículo híbrido enchufable. El principal mérito de la solución propuesta es la validación del método en un prototipo en tiempo real, que según se verá en la revisión del estado del arte, es una de las principales carencias en la mayoría de estudios de esta técnica en automoción. Para llegar a esta solución, el desarrollo de un modelo del sistema adecuado y la definición del problema de optimización en combinación con el uso de una herramienta numérica fiable y robusta, han sido imprescindibles y por eso ocuparán una parte importante de este documento. Asimismo la validación por medio de simulación previa a la experimental, así como el diseño de unas condiciones de conducción repetibles, para comparar el control propuesto con el original del vehículo, serán tratadas antes de llegar a la validación y discusión finales.

Resum (catalan)

Diversos factors socioeconòmics estan portant als governs a fomentar els vehicles propulsats elèctricament. Actualment, el coll d'ampolla per a la producció en massa del vehicle elèctric resideix a la tecnologia de la bateria d'alta tensió. Un dels reptes principals per assegurar les prestacions de seguretat, confort, funcionament i durabilitat de la bateria és la gestió tèrmica, ja que a temperatures allunyades de les especificades pel fabricant, aquesta envelleix de forma prematura, donar lloc a una perillosa i descontrolada reacció exotèrmica i/o ser incapaç de lliurar l'energia elèctrica necessària per moure el vehicle. La tendència en el disseny de solucions per a la gestió tèrmica, és la d'usar circuits de refrigeració amb arquitectures cada vegada més sofisticades que impliquen la necessitat d'un major nombre d'actuadors elèctrics com a bombes, ventiladors i electrovàlvules. El control d'aquests sistemes és complex a causa del seu comportament no lineal, a l'elevat nombre d'entrades i sortides i a la necessitat de assolir diversos objectius alhora sovint contradictoris. Davant aquesta classe de problemes, els mètodes convencionals de control són portats al seu límit i nous mètodes basats en optimització, com el control predictiu, capaços d'explotar el potencial d'aquest tipus de sistemes, comencen a atreure l'atenció del sector.

Aquesta tesi tracta del disseny d'un control predictiu per a la gestió tèrmica del circuit de refrigeració de la bateria i l'electrònica de potència en un vehicle híbrid endollable. El principal mèrit de la solució proposada és la validació del mètode en un prototip en temps real, que segons es veurà en la revisió de l'estat de l'art, és una de les principals manques en la majoria d'estudis d'aquesta tècnica en automoció. Per arribar a aquesta solució, el desenvolupament d'un model del sistema adequat i la definició del problema d'optimització en combinació amb l'ús d'una eina numèrica fiable i robusta, han estat imprescindibles i per això ocuparan una part important d'aquest document. Així mateix la validació per mitjà de simulació prèvia a l'experimental, així com el disseny d'unes condicions de conducció repetibles, per comparar el control proposat amb l'original del vehicle, seran tractades abans d'arribar a la validació i discussió finals.

Abstract (german)

Elektrisch angetriebene Fahrzeuge werden derzeit aus unterschiedlichen sozioökonomischen Gründen von den Regierungen gefördert. Der Flaschenhals für die Massenproduktion von dieser Technologie ist die Hochvoltbatterie, die mehrere Herausforderungen um die Sicherheit, Komfort, Performance und Lebensdauer Anforderungen zu erfüllen begegnet. Das Thermomanagement ist eine davon, da der Batterie Temperaturbetrieb außerhalb des vom Hersteller angegebenen Bereiches führ zur vorzeitigen Alterung, gefährlichen und unkontrollierten exothermischen Reaktionen und Stromversorgungsbeschränkungen, die den Antrieb des Fahrzeugs verhindern. Der Trend in das Design von Thermomanagementlösungen geht dahin, immer ausgeklügelter Kühlkreislaufarchitekturen zu verwenden, wobei eine steigende Anzahl von elektrischen Aktoren wie Pumpen, Lüftern und Ventilen, benötigt werden. Die Regelung solcher Systeme ist komplex aufgrund ihres nichtlinearen Verhaltens, der zahlreichen Eingängen und Ausgängen und des Bedürfnisses mehrere Ziele gleichzeitig zu erfüllen. Angesichts dieser Aufgabenstellung, werden die konventionelle Regelungsmethoden an ihre Grenzen gebracht und neue optimierungsbasierten Methoden wie die modellbasierte prädiktive Regelung, die das volle Potential solcher Systeme ausnutzen können, beginnen die Aufmerksamkeit der Autoindustrie auf sich zu lenken.
Die vorliegende Arbeit handelt von dem Design einer modellbasierten prädiktiven Regelung für das Thermomanagement des Batterie- und Leistungselektronikkühlkreislaufs in einem Plug-In Hybrid Fahrzeug. Der Hauptverdienst der vorgeschlagenen Lösung ist die Methodenvalidierung in einem Prototyp in Echtzeit, was häufig, wie in der Stand der Technik angeführt werden wird, einer der Mangeln der meisten Veröffentlichungen in dieser Branche ist. Um zu dieser Lösung zu gelangen, eine geeignete Modellierung und Optimierungsproblembeschreibung sowie ein effizientes und robustes numerisches Werkzeug waren wesentlich und spielen deshalb eine wichtige Rolle in diesem Dokument. Weitere angegangene Punkte bevor der Versuchsauswertung und Schlussfolgerungen, sind die Simulationsvalidierung und das Entwurf von wiederholbaren Fahrbedingungen für einen gültigen Vergleich der vorgeschlagenen Regelung gegenüber der ursprünglichen im Fahrzeug.

Acknowledgements

The present thesis has been developed within the Industrial Doctorates Plan directed by the Catalan Government, Generalitat de Catalunya, which promotes companies and universities collaboration to contribute to the competitiveness and internationalization of the Catalan Industry. Through this, as a PhD student I had the great opportunity of combining the academic supervision of the thesis by joining an experts research group at the Polytechnic University of Catalonia (UPC) called CREMIT, with experience in the car industry mentored by the manufacturer SEAT, member of the Volkswagen Group. Thanks to this initiative driven by several actors, it has been possible for me to spend nearly two years abroad, at the VW group headquarter in Wolfsburg (Germany), attend to several courses, workshops, test drives and conferences, thus acquiring all the necessary contacts, technical skills and tools to design this project. Therefore I am very satisfied with the enriching experience along these three years and feel a duty to thank, among a long list, the following people:

To Gerhard for proposing me this journey and accompany me along it.
To Jesus, Manolo and Rafa for forming such a friendly team and accepting me in it.
To Carlos O. for joining us with his time, jokes and predictive experience.
To Tino F. for believing in me.
To Tobias and Frank for their patronage and optimism.
To TLK-Thermo GmbH and Michael for providing me with their wonderful simulation tools.
To Wenzel, Jan, Ulf, Tino K., Robert and Christian for their disinterested support.
To Rene for training me.
To Martin for his patient support behind the scene.
To Manuel G. for pointing me the way with his ideas.
To Julian for his flexible and valuable support.
To Mario for protecting me when I was scared of the car.
To Sergio, Juanjo and Marc for their time and professionalism.
To Jordi A. for humanizing bureaucracy.
To Adri for always replying "yes".
To Manuel T. and Jonathan for the inspiring walks around the lake.

To my family and friends for their interest and ears.
To my parents, Lourdes and Jorge, brother, Edu, and especially my fiancée, Alejandra, for living with me the daily triumphs and shortcomings of this thesis.

Contents

List of Figures

List of Tables

Abbreviations

AC Air Conditioning
ACC Adaptive Cruise Control
BAT Battery
BEV Battery Electric Vehicle
CAN Controller Area Network
COP Coefficient Of Performance
DAE Differential Algebraic Equation
DMS Direct Multiple Shooting
DP Dynamic Programming
DSS Direct Single Shooting
ECU Electronic Control Unit
EM Electric Machine
EMPA Swiss Federal Laboratories for Materials Science and Technology
E-Mobility Electromobility
ESC Electronic Stability Control
ETH Swiss Federal Institutes of Technology
ETK Emulator test probe
EV Electric Vehicle
FCHEV Fuel Cell Hybrid Electric Vehicle
FMI Functional Mock-up Interface
FMU Functional Mock-up Unit
FPGA Field Programmable Gate Array
GPS Global Positioning System
GUI Graphical Users Interface
HEV Hybrid Electric Vehicle
HIL Hardware In The Loop
HV High Voltage
HVAC Heating, Ventilating, and Air Conditioning
ICE Internal Combustion Engine
IGBT Insulated-gate Bipolar Transistor

IVP Initial Value Problem
KKT Karush-Kuhn-Tucker
LP Linear Program
LV Low Voltage
MIOCP Mixed-Integer Optimal Control Problem
MIMO Multiple Inputs and Multiple Outputs
MPC Model Predictive Control
NEDC New European Driving Cycle
NLP Nonlinear Program
NMPC Nonlinear Model Predictive Control
NTU Number of Transfer Units
OCP Optimal Control Problem
ODE Ordinary Differential Equation
OO Object Oriented
PE Power Electronics
PHEV Plug-in Hybrid Electric Vehicle
PID Proportional–Integral–Derivative
PTC Positive Temperature Coefficient
PWM Pulse Width Modulation
QP Quadratic Program
REEV Range-Extended Electric Vehicle
RP Rapid Prototyping
RTI Real-Time Iteration
SCR Selective Catalytic Reduction
SIL Software In The Loop
SOC State Of Charge
SQP Sequential Quadratic Programming
TM Thermal Management
TTW Tank-to-Wheel
USB Universal Serial Bus
WTW Well-to-Wheel

Symbols

M	Torque	N m
n	Rotational Speed	rpm
P	Power	W
T	Temperature	°C
v	Linear speed	$\frac{\mathrm{km}}{\mathrm{h}}$
$\dot{V}$	Volume flow rate	$\frac{\mathrm{L}}{\mathrm{min}}$
$\dot{m}$	Mass flow rate	$\frac{\mathrm{kg}}{\mathrm{s}}$
$\dot{W}$	Heat flow rate	W
I	Electrical current	A
m	Mass	kg
c	Specific heat capacity	$\frac{\mathrm{J\,K}}{\mathrm{kg}}$
A	Surface	m^2
h	Heat capacity rate	$\frac{\mathrm{W}}{\mathrm{K}}$
U_i	Internal energy of the system	W
l_m	Characteristic length	meter
Re	Reynolds number	1
Nu	Nusselt number	1
Pr	Prandtl number	1
U	Voltage	V
E	Energy	J
Ri	Internal Resistance	Ω
τ	Time constant	1
ρ	Density	$\mathrm{kg/m}^3$
ϵ	Emissivity	1
α	Convection coefficient	$\mathrm{W/(m^2K)}$
ϑ	Kinematic viscosity	$\mathrm{cm}^2/\mathrm{s}^{-1}$
λ	Thermal conductivity	W/(mK)
μ_x	Mean of x	Same units as x
σ_x	Standard deviation of x	Same units as x

Dedicated to my father

Chapter 1

Introduction

"Words are like eyeglasses they blur
everything that they do not make clear."

— Joseph Joubert

CONTENTS:

This chapter leads to the reader through the motivation for the present research. First, the problem that motivated the research is presented, followed by the solution proposed in this thesis and the state of the art in this field. Finally, the specific objectives of the thesis as well as the thesis's outline are described. After reading this chapter, the reader will wear "glasses" with which each word in the title of this thesis is clear.

1.1 Motivation and previous concepts

1.1.1 Electromobility typologies

Electromobility (E-Mobility) is a general term for the development of electric-powered drive-trains. Since numerous vehicle typologies belong to this classification, an overview of them is helpful:

- **Battery Electric Vehicle (BEV)**: A BEV uses chargeable high voltage batteries as unique energy source to power one or more electric motors responsible for the propulsion.
- **Range-Extended Electric Vehicle (REEV)**: A REEV works exactly as a BEV in the sense that it uses a chargeable battery to feed an electric motor but includes another energy source: a fuel tank to extend vehicle driving range once the battery is empty. An Internal Combustion Engine (ICE) and an electric alternator are needed to transform the chemical energy contained in the fuel to electrical energy. The propulsion of the vehicle is still purely electric, the ICE is only used to charge the battery.
- **Hybrid Electric Vehicle (HEV)**: An HEV has two different energy sources: a non-externally chargeable battery and a fuel tank and combines a conventional ICE with an electric motor for the propulsion. HEVs can be further classified depending on the level of hybridization:
  - **Micro Hybrid**: A micro hybrid is a conventional vehicle with a bigger alternator to allow the start-stop mode.
  - **Mild Hybrid**: A mild hybrid has an electric motor which is not able to power the vehicle alone. It supports the ICE and is used for regenerative braking.
  - **Full Hybrid**: A full hybrid has an electric motor able to power the vehicle alone.

  and the drive train structure:

  - **Parallel Hybrid**: A parallel hybrid has a structure where both the electric motor and the ICE have mechanical connection to the drive shaft.
  - **Series Hybrid**: A series hybrid has a structure where only the electric motor has mechanical connection to the drive shaft. The ICE is used to charge the battery.
- **Plug-in Hybrid Electric Vehicle (PHEV)**: A PHEV has two different energy sources: an externally chargeable battery and a fuel tank and combines a conventional ICE with an electric motor for the propulsion.

- **Fuel Cell Hybrid Electric Vehicle (FCHEV)**: A FCHEV converts chemical- to mechanical energy burning, e.g., hydrogen in an ICE or making it react with oxygen in a fuel cell to power an electric motor.

In the rest of this thesis, the term Electric Vehicle (EV) will be taken for all types of electrified vehicles shown in this classification.

1.1.2 E-Mobility roadmap

Despite the internal combustion engines predominance in personal transportation in the last century, they are called to loose the leading role due to EVs present numerous advantages in comparison:

- **Lower climate impact**: The combustion of fuel produces exhaust gases that cause Global Warming such as carbon dioxide CO_2 and nitrous oxide NO_x. On the contrary, Well-to-Wheel (WTW) studies show that in general vehicles reduce these greenhouse gases emissions with increased electrification. Note that the improvement depends on the fossil content of the electricity mix [1].
- **Higher efficiency**: Electric drivetrains are more efficient than the ICE because of their high-energy conversion efficiency from Tank-to-Wheel (TTW) that includes no idling losses, braking energy recuperation or the possibility of not using a gearbox.
- **Lower pollution**: In conventional vehicles the chemical to mechanical energy transformation is done in the vehicle itself and the emission of pollutants with adverse health effects like NO_x or Diesel particles remain in the close environment. Instead, in EVs the final electrical to mechanical energy conversion is done with no local tail pipe emissions, thus not contaminating the close environment. This geographical aspect of emissions is not to be underestimated since future world demography is supposed to be concentrated even more in big cites [2].

All the mentioned drawbacks have led governments to push the transition to E-Mobility. This is materialized on the one hand in the increasing stringent emissions standards that all new vehicles have to pass for registration as it can be seen in figure 1.1:

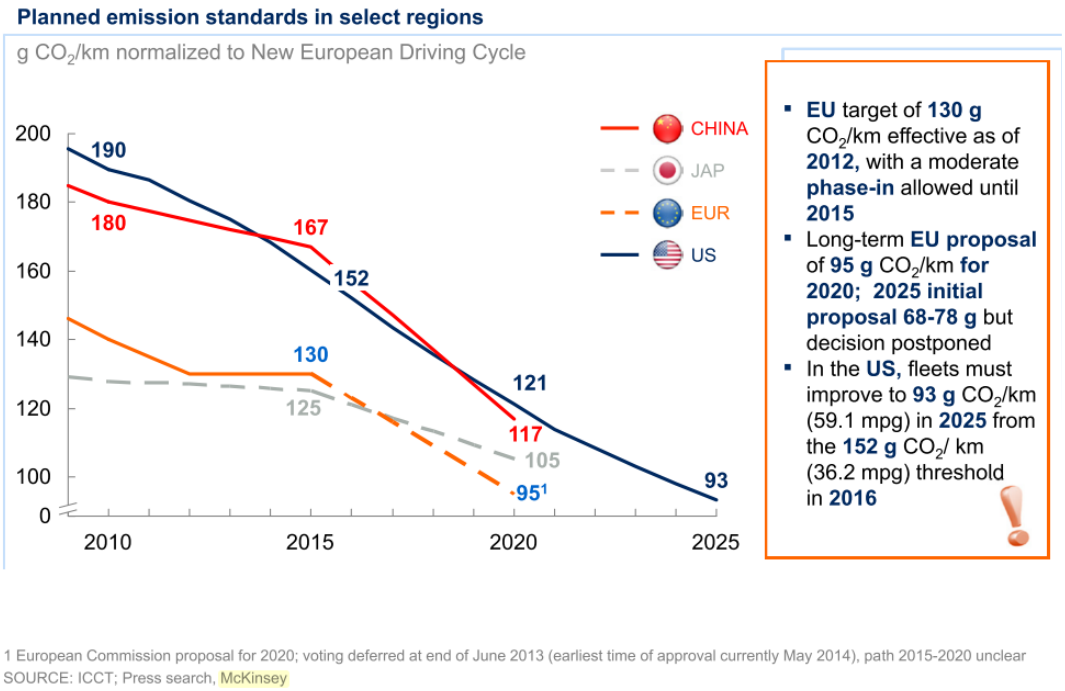


FIGURE 1.1: Planned emission standards in select regions [3].

On the other hand, governments are trying to stimulate consumers demand with subsidies for purchasing EVs, as shows figure 1.2.

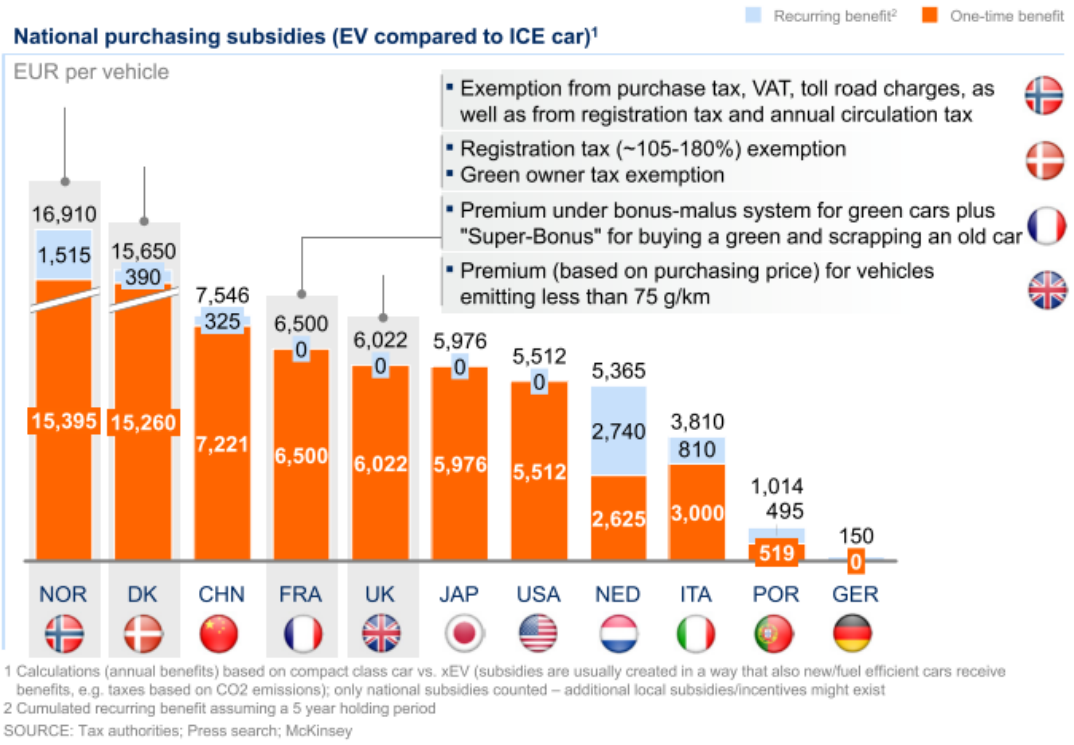


FIGURE 1.2: National purchasing EVs subsides [3].

Nevertheless, the road to emissions free transport in future cities has still some hurdles to overcome related to technical and socioeconomic issues, as shown in figure 1.3:

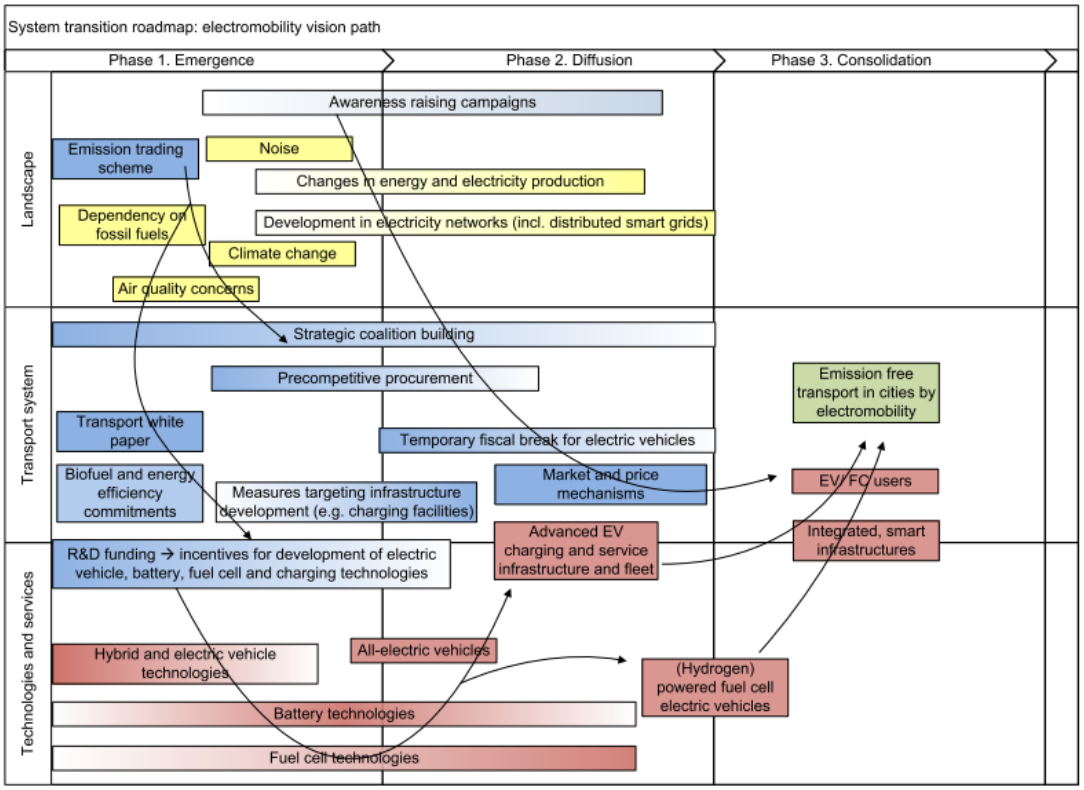


FIGURE 1.3: E-Mobility roadmap [4].

1.1.3 Thermal Management for E-Mobility

Despite the many advantages of EVs, there are many technological challenges still open that require a solution. Among them, the battery-related are the most critical, as it can be seen in the red box on the bottom of figure 1.3. The state of the art chemistry in High Voltage (HV) batteries for transportation applications is the lithium (Li) ion technology [5–7] whose main features are summarized in Table 1.1. As it can be seen in the table, the specific energy stored in a Battery (BAT) is limited. One kilogram gasoline contains above 12000 Wh, more than fifty times the energy in a kilogram Li-ion BAT. This fact leads to electric autonomies that are limited in general to a few hundred kilometers in mass-produced vehicles with popular prices. Nevertheless, large effort is currently being put in offering electric vehicles with considerable autonomies at attractive prices. In this scenario, the American auto-maker Tesla has recently become the spearhead of affordable E-mobility launching its Model 3 with around 340 km autonomy at the price of 35000$ [8].

On the other hand, batteries age over time due to several internal degradation mechanisms [9] that are accelerated with usage. In this sense, battery manufacturers give details with the cycle life and lifetime in Table 1.1. The cycle life gives the number of discharge/charge cycles a BAT can experience before it fails to meet specific performance criteria, for example before its capacity falls under 80% of its original value. The lifetime is the period of time before a BAT fails to meet specific performance criteria, whether by active or inactive use and it is given in years.

TABLE 1.1: Li-ion batteries performance for the short, medium and long term [10].

	2015	2025	beyond 2025
Specific energy (Wh/kg)	150	200	250
Specific power (W/kg)	400	500	500
Efficiency (%)	90	92	95
Cycle life (# cycles) @ DoD	1000 to 3000 with 80%		
Lifetime (years)	7	10	12
Operating temperature	improving but uncertain		
Safety	good		
Cost ($/kWh)	400-600	300-400	250-300

Moreover, at high temperatures Li-ion batteries ageing mechanisms are accelerated [11] and above certain threshold, typically 60°C, they can even lead to dangerous uncontrolled exothermic reactions in the BAT pack [12]. However, extreme cold temperatures are neither desired since they increase significantly the electrical impedance of the cells becoming predominant bellow -10°C [13]. As a consequence, the BAT in an EV could even be incapable of supplying the necessary current for starting the vehicle in a cold winter day.
For these reasons, batteries for electric propulsion should work within the temperature range specified by the manufacturer(typically 20°C-30°C) [14, 15]. This range is clearly narrower than the vehicle operation window, which has to assure extreme hot ($\approx$ 60°C) and cold climates ($\approx$ -25°C). Analogously to the BAT, the Power Electronics (PE) module, which is responsible for the control of the Electric Machine (EM), ages under thermal stress. Although the costs of replacing this component are lower than in the BAT case, a failure in the PE due to high temperatures should be avoided as well [16].

Given the effect of temperature on the electrical components of a PHEV, as the BAT and the PE, the study of Thermal Management (TM) is crucial for safety, performance and durability requirements. The general concept of TM covers the heat transfer in the whole vehicle: in

the components and the cabin. This makes EV TM an interesting research field with several challenges, as evidenced in Table 1.2.

Concerning the cabin comfort, EVs generate considerably less heat compared to ICEs. For this reason, at cold temperatures, a heat pump [17] or/and a Positive Temperature Coefficient (PTC) semiconductor heater [18] are necessary for heating the cabin. Both solutions consume considerable electrical energy thus reducing the electric range of the vehicle.
From the components point of view, the main challenge is to keep them within the optimal temperature range by means of cooling or heating processes. In the case of conventional vehicles, the TM is necessary to achieve temperatures that reduce greenhouse gases and pollutants emissions, friction losses and assure safety operation.
In EVs the TM becomes even more critical, because besides dealing with components that are more temperature-sensitive, EVs compared to conventional vehicles dispose of less resources for cooling and heating them: first they lack the heat rejected from combustion and second, the electrical consumption, needed for moving the cooling/heating actuators, has a higher impact in the vehicle driving range.

1.2 Research topic

In order to comply with the increasing number of requirements in EVs TM several solutions exist [19]. One key aspect is the used transfer medium, typically air or liquid, and the usage or lack of active heating/cooling elements such as evaporators, heating cores, engine coolant, or even electric and fuel-fired heaters. Based on the combination of these options, the authors in [19] show several solutions with air, figure 1.4, and liquid, figure 1.5, for TM. Finally, they conclude that a TM system using air as the heat transfer medium is less complicated, though less effective, than a system using liquid cooling/heating.

In other words, the more cooling/heating options a system offers, the more accurate the resulting temperature control is and the more actuators are required to perform the heat transfer along the complex pipes architectures and through the several heat exchangers.

The high number of actuators needed for an accurate TM presents a big challenge in several aspects. First, complex systems are characterized by their high nonlinear behaviour and in such scenario, the more degrees of freedom (introduced by the number of actuators), the more difficult becomes the problem of finding the best controls needed for an effective TM. Moreover, nowadays most actuators used for cooling/heating such as pumps, valves and fans are electrical. Being the limited vehicle autonomy a critical aspect in EVs, electrical

consumption for purposes other than driving is to be avoided as far as possible.
Therefore, for EVs is particularly crucial not only to achieve an accurate TM but also to do it in an efficient way, spending as less electrical energy as possible for this purpose.

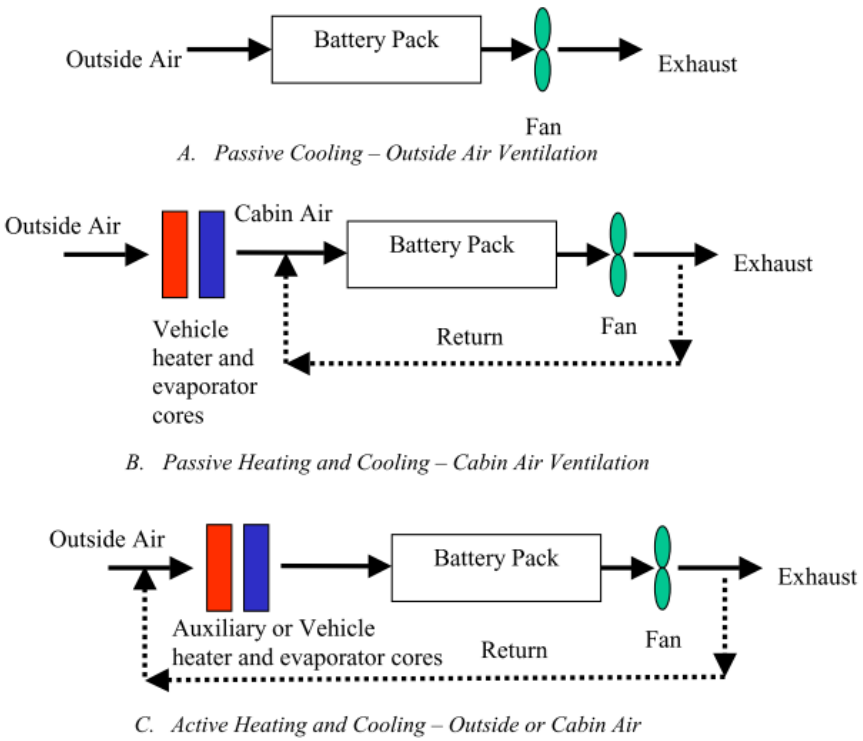
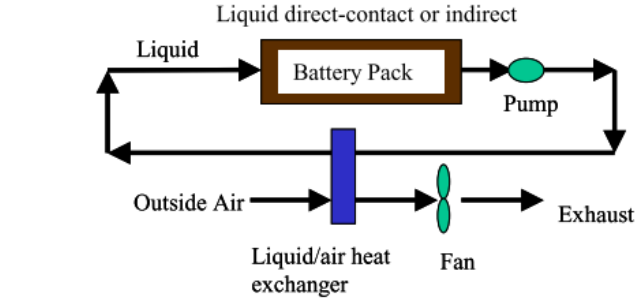


FIGURE 1.4: TM solutions taken from [19] with air as heat transfer medium.

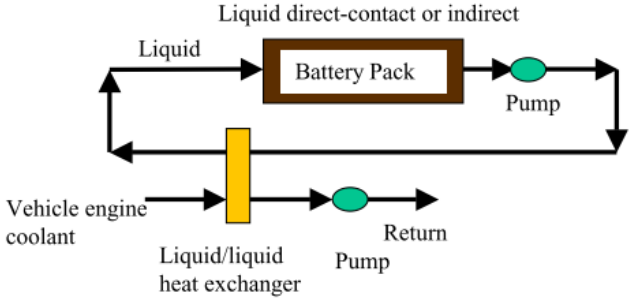
The challenge of achieving an accurate and efficient TM for a mobile application is specially complex due to the highly transient working conditions: the pedal demand of the driver, the traffic, the road, the climate and many more variables have a considerable effect on the thermal load and the cooling/heating resources available. Hence, the problem of finding the best controls for achieving an accurate and efficient TM is a complex issue in EVs which is claiming for new solutions.

A classical approach for finding the "best" possible controls is to use a bunch of rules based on experience that aim at characterizing the vehicle behaviour under all possible conditions with a limited number of scenarios. For example, the use of Proportional–Integral–Derivative (PID) controllers requires a previous tuning that once done remains as a closed and static box that has to face the diverse situations hopefully projected during the tuning.
These strategies work and guarantee the safe operation of the vehicle. Nevertheless, the more complex systems are, the lower is the possibility of identifying the best controls for the

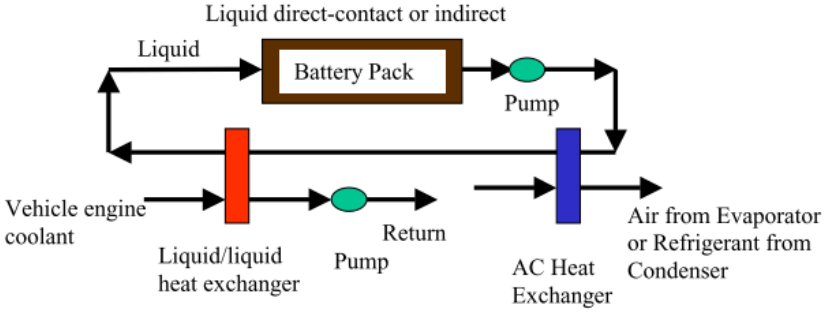
diverse driving situations and conditions and the larger is the potential lost.



D. Passive Cooling – Liquid Circulation



E. Active Moderate Cooling/Heating – Liquid Circulation



F. Active Cooling and Heating – Liquid Circulation

FIGURE 1.5: TM solutions taken from [19] with liquid as heat transfer medium.

The opportunity of exploiting all this unreleased potential has opened the door for methods under the umbrella of the optimal control family. Among them, the study of MPC is particularly interesting for optimizing high nonlinear processes with Multiple Inputs and Multiple Outputs (MIMO) and several goals such as the ones governing the TM for EVs.

Just to get an overview of the idea behind MPC which will explained later, it can be said that it consists of:

- a model based on an Ordinary Differential Equations (ODEs) system or/and data-based look-up tables
- an ODEs solver
- a set of goals and constraints describing the desired performance of the controlled plant
- a numerical optimization algorithm

With these elements, MPC can find the best possible control for a certain prediction horizon. It just needs to know the actual state of the plant to perform the following iterative steps:

- given a candidate for the control signals to be applied along the prediction horizon, solve an Initial Value Problem (IVP) to **obtain a predicted trajectory** for the controlled plant state
- use the controls candidate and states trajectory prediction to **evaluate the** "**success**" of the control according to the fulfilment of goals and the violation of constraints and decide if it is acceptable enough to stop the search or not
- if required, generate control candidates that improve the results obtained

The advantage of using a model to predict the future scenario in comparison to a classical method such as the PID can be seen in figure 1.6.

Another interesting feature of MPC is that it can easily exploit future driving information to improve the model predictions and thus obtain better results. Given an online predictor of the driving cycle for the next kilometers, like the one developed in [21], an MPC approach for TM would be able to improve the use of the resources to cool/heat the components. For example, being in a highway and knowing that in the next kilometers a slope or a traffic jam will be faced, the MPC controller, taking into account the coming thermal load, could start cooling the components, even if at the current moment they were at benign temperatures. This way, the high air mass flow entering through the cooler of the vehicle due to the high speed could be used to pre-cool the components in advance, while without the future information, the need of cooling them would be addressed once in the traffic jam where due to the low speeds only a low amount of air mass flow would be available.

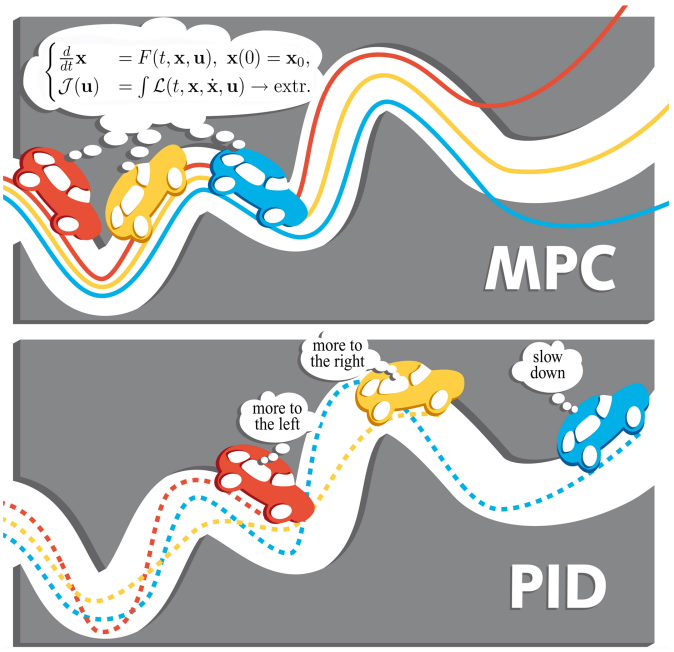


FIGURE 1.6: MPC vs PID taken from [20]

The aim of this dissertation is to contribute in the field of TM for EVs generating knowledge related to new optimal control methods. Our proposal is to validate the potential of optimal control methods for TM not only by means of simulations, but to go a further step and gain experience of its implementation in the vehicle. Essential steps for these ambitious goals are to develop a valid model of the studied system, characterize the control goals and constraints, validate the controller through simulation results and finally to roll the sleeves up and solve all possible setbacks to obtain experimental results in a real vehicle.

1.3 Research problem

Given the high temperature sensitivity of the electric components in EVs and the limited electric energy available for driving, TM strategies are crucial to match their safe, accurate and efficient performance. Since first series EVs are just slowly emerging in the market, TM main efforts have been put in the design of the cooling package and its effective control to assure safety operation. Nevertheless, there is a lack of research with regard to optimal control methods for achieving not only a functional but also an efficient TM able to exploit all possibilities of complex dynamic systems.

In order to achieve an accurate and efficient TM in an EV, three main targets have been defined for this study.

- **Flexible MPC development:** the aim here is to build a model of a specific controlled plant and to formulate its goals and restrictions in a systematic way that can be easily modified and extended for other TM architectures and to incorporate further features such as a driving cycle prediction.
- **Build of simulation environment for validation:** the goal is to profit from simulation tools with which time and money consuming experiments can be spared. A key issue in MPC is the scaling and initialization of the problem, if the optimizer is started far from the optimal results, it will be more difficult for it to get there. With a good simulation environment, knowledge of the process which is crucial for the optimizer settings can be generated.
- **Real time MPC:** the most ambitious target of this research is to obtain a real-time implementation of an MPC in a vehicle. Computational burden is one of the main hurdles for MPC broad spread in the automotive sector and one of the aims of this thesis is to contribute in the overcoming of these hurdles.

1.4 State of the art

Due to the novelty of this research topic, there are not many studies dealing with MPC for TM in EVs in the literature. For this reason, this sources review pretends to be a guideline to understand the necessity of researching in this field. With this aim the state of the art was divided in several subsections.

1.4.1 Brief situation: MPC origins

MPC found its first applications in the process industry, around the mid-70s. Petrochemical plants and refineries presented the perfect scenario enhancing MPC features: on the one hand the dimensions of the plants (with thousands of states and many controls) suggested an unreleased potential not exploited by "manual" tuning while on the other hand, the slow inertia of the systems dynamics enabled real time computations. During the second decade which started around late 80's, the increasing industrial development of MPC software, which included the addition of constraint handling, was coupled with research studies dealing with robustness or stability that gave scientific trust to the industrial method and placed MPC among some of the best known optimal control techniques. In the last decade new concepts

related to other sectors demands were started to be addressed, applications with inherit orders-of-magnitude smaller inertia, high non linearity and/or presence of control variables which only can assume integer values claimed for more powerful algorithms. This is the preamble to current MPC for automotive applications, for a deeper insight, see [22, 23].

1.4.2 MPC/NMPC automotive applications

In this thesis the nonlinear variant of MPC, Nonlinear Model Predictive Control (NMPC), has been used. The main difference is that it relies on a nonlinear model to build the predictions. The advantage of NMPC is that in general most processes in the real-world are nonlinear and hence, the predictions obtained with a nonlinear model are more realistic. Nevertheless, in contrast to MPC problems, the nonlinear counterpart is in general, non-convex which demands more complex numerical methods and it is a challenge for stability theory [24].

Fortunately, in recent years faster processors with more memory space opened the door to new algorithms for fast MPC/NMPC applications. This has resulted in a considerable growth of the number of applications in the automotive sector, where MPC/NMPC has been applied. In [25–27] predictive control is used to improve the Adaptive Cruise Control (ACC) of a vehicle in terms of safety, avoiding collisions; and efficiency, skipping unnecessary accelerations that lead to extra emissions. The latter goal is also pursued in [28–30] where NMPC is used to improve the performance of Diesel motors overtaking the control of the Selective Catalytic Reduction (SCR) or of the engine air path. Much attention has been put in the energy management of hybrid vehicles, where the aim is to manage the torque split of the different machines optimally [31–35]. Another field where MPC has shown its potential is the fuel cell, where the control of the air-supply is crucial to guarantee safety and to achieve a high performance [36]. [37] uses NMPC for calculating the optimal trajectory in a truck guaranteeing safety and driving comfort and [38] exploits the MPC potential in the Electronic Stability Control (ESC).
Although these applications and many more show the increasing interest to study MPC/NMPC for the optimal control of automotive systems, there still remains much to be done for the decisive step from research to development. This is reflected in the following limitations of many existing applications:

- **Real vehicle implementation:** despite the final goal is to perform the NMPC in the real plant, the vehicle, with its transient conditions, many studies validate the control by means of simulations, usually relying on a high-fidelity model of the plant, [25,

27, 28, 30, 31, 35, 37–41] or in a test bench [36, 39]. This first validation is necessary to prematurely identify errors thus avoiding future expensive tests repetitions. Nevertheless, to achieve real vehicle implementation a large effort is still required.

- **Real time performance:** since many studies evaluate NMPC with simulations, not all of them concentrate on computation time, putting much effort first in the method potential for the applied field. Nevertheless, in the last years this has changed and many papers are focused on real time performance [36, 37, 42].

- **Small problem:** Since the number of controls influences directly the size of the optimal control problem, to avoid high computational times, most NMPC applications aim at solving problems with no more than three control signals governing the plant [27, 28, 31–35, 39, 41, 43]. Although this scenarios are suitable to test NMPC potential, real vehicle systems usually present many control variables.

- **Future information:** as mentioned before, the use of available future information of the trip is a key feature of NMPC for developing its entire potential in the vehicle. Nevertheless, since systems predicting the future vehicle behaviour are just in an early implementation stage, less studies add to the complexity of the NMPC problem itself, the generation of future driving conditions. An example of NMPC with future information exploitation is shown in [26].

1.4.3 MPC/NMPC automotive applications for TM

Although the TM of ICEs is interesting from the efficiency and safety point of view and some NMPC applications exist [39, 41], the challenge of finding the optimal control for the cooling/Air Conditioning (AC) circuits is not as critical as in EVs, where the ICE as heat source is smaller or non-existent, the electrical consumption has a tremendous impact on the vehicle autonomy and in addition, the electrical components are highly temperature sensitive.

Since E-Mobility is not in the consolidation phase of figure 1.3 yet, electric components final solution concerning cost, safety, performance, durability and other requirements is not established by the moment. For example, in the case of the BAT, the core of the EV, although the Li-ion is called to be the leading technology in the coming years, there is still numerous research on other materials that could offer better performance in range, cost or durability terms [10]. Another example are the different existing options for the electrical machine, as shown in figure 1.7:

	IM[a]	DCM[b]	SRM[c]	PMSM[d]	SyncRM[e]	AFM & TFM[f]
Weight and volume			Medium	Good	Medium	Good
Energy efficiency	Medium		Medium	Good	Medium	Good
Fault tolerance, Robustness	Good		Good	Medium	Medium	Medium
Cost	Good	Good	Medium		Medium	
Recycling	Good	Good	Good		Good	Medium

[a] Induction motor, [b] DC motor, [c] Switched reluctance motor, [d] Permanent magnet synchronous motor, [e] Synchronous reluctance motors, [f] Axial flux motor & Transverse flux motor

FIGURE 1.7: Different alternatives for the electric machine taken from [1].

Given that the electric components for mass-production are not fixed yet, consequently the challenge of keeping them under benignant temperatures has been focused on the TM architecture and the selection of the heat transfer medium [19, 44, 45]. For this reason few literature concerning TM control in EVs is available and even less if the focus is put on optimal control methods.
However, in [40] we find an MPC application that exploits its multi-objective and MIMO nature for the optimal cabin heat TM in a HEV while the authors of [46], with who we collaborated during the elaboration of this thesis, applied NMPC in the power and thermal management of an HEV.

1.4.4 Novelties and contributions of this thesis

Regarding the state of the art, the aim of this paper is to contribute to the existing research with a **new** NMPC application for the TM in a PHEV, that distinguishes itself from others mainly in the following points:

- it deals with **a high number of control signals**, six in total
- it is performed **on real time in a vehicle**

Hopefully, this effort does its bit in the implementation of NMPC in the future vehicle. The road map to NMPC embedded in the real vehicle can be seen in figure 1.8, where the final goal is to have the algorithm running embedded in the vehicle. In this sense, some researches point Field Programmable Gate Array (FPGA) or multicore microprocessors as the suitable platforms to exploit parallelization of the NMPC controller design [47].
Before the final vehicle implementation is achieved, NMPC simulations can be used in the

offline and online variants, as shown in the first two steps in figure 1.8, to gain trust as a reliable control methodology. Furthermore, the simulation results can be used to identify new operating points and to improve the conventional rules-based control strategies.
Hardware In The Loop (HIL) tests can be useful, as well, to generate more knowledge and improve the NMPC implementation.
However, the crucial validation takes places in the real vehicle. Therefore, this thesis deals with the first step in the vehicle implementation, as shown with the red arrow in figure 1.8, which consists of using a Rapid Prototyping (RP) module and a Laptop to perform the NMPC.

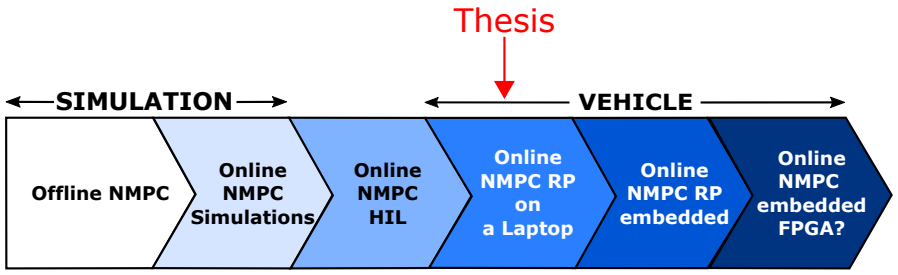


FIGURE 1.8: NMPC road map in the automotive sector.

1.5 Hypothesis

In order to complete this research within the duration of three years and reach the goal of implementing a NMPC strategy capable of real-time performance in a vehicle, the following hypotheses were taken:

- Despite the vehicle used in this research is a PHEV, only the pure electric driving mode was studied. The main reason for this decision was that the components in the vehicle are cooled in three different complex circuits and to model and control all of them on real-time would have been an excessive task for a first NMPC application. On the other hand, the electric mode is very interesting since it is supposed to be the one selected for the day-to-day trips.
- No future information concerning the driving cycle to be performed by the vehicle is used. Instead it is assumed that the speed and other relevant mechanical inputs of the model are kept constant within the prediction horizon.
- The optimization algorithm used for real-time NMPC belongs to an existing optimization package called MUSCOD-II [48], developed by the Interdisciplinary Center for Scientific Computing (IWR) of the Heidelberg University.

- The studied TM control problem deals with three solenoid valves that only accept binary values. In our approach they are defined in the optimizer as continuous variables with 0 and 1 as upper and lower bounds and the optimal values found are then rounded before being sent to the actuators. The suitable approach would have been to define them as binary variables. Nevertheless, the methods required for the solution of mixed-integer systems are, in general, more complex and imply more computational burden, specially in case of a high number of control signals like is the case in this research. To achieve a real-time capable NMPC for TM the rounding simplification was taken.
- The coolant was modelled as an incompressible fluid and the surface in contact between coolant and the heat source in the components is assumed to be large enough to ensure a fast equilibrium of the coolant outlet and the component temperature.

1.6 Aims and objectives

The different aims and objectives pursued in this research are:

- Build a model of the cooling circuit capable of describing its multidisciplinary behaviour with a trade-off between accuracy and simplicity.
- Validate the model with experimental data.
- Formulate the Optimal Control Problem (OCP) describing the goals and constraints needed for a successful TM.
- Validate the NMPC performance by means of simulations using as controlled plant the same model contained in the NMPC controller.
- Build a model that describes a classical control approach for TM.
- Compare the results obtained with NMPC and the classical control approach in several driving scenarios that permit to obtain an overview of the predictive method strengths and weaknesses.
- Design a tool chain that enables the bypass of the original controls to take the NMPC output instead for the cooling circuit in a PHEV prototype.
- Design a repeatable driving cycle suitable for comparison.
- Perform different experimental test drives under comparable conditions once with the NMPC and once with the original vehicle controls and validate the results.

1.7 Thesis outline

This thesis has been divided in seven chapters that will describe the several steps done till the achievement of the mentioned objectives. For supporting the essential information contained in the chapters, two appendixes are included at the end of this dissertation.

In **chapter 1**, an introduction to the research topic and the NMPC state of the art in the automotive sector are presented.
In **chapter 2**, the controlled plant for which the NMPC will be studied is described.
In **chapter 3**, the main multi-domain equations behind the developed model of the controlled plant are commented. This is supported with the code of the model, found in Appendix A.
In **chapter 4**, the OCP formulation for the PHEV TM is presented. Besides these details, the numerical method for solving the OCP used in MUSCOD-II is explained. For a better understanding in how the optimizer was used, Appendix B contains an example of the code required from the user.
In **chapter 5**, it is described how the concatenation of OCP, which is the basis of NMPC, is done. Furthermore, the controlled plant for simulation, the classical control model and the simulation results are shown and discussed.
In **chapter 6**, the NMPC validation is addressed in the PHEV prototype. Before comparing the results obtained with the original vehicle and the NMPC controller, the repeatable test drives designed are shown.
And finally, in **chapter 7**, the conclusions and future works suggestions bring the thesis to a close.

TABLE 1.2: TM requirements in EVs and conventional cars.

Thermal Managent requirements		Conventional car	Electric Vehicle
Cabin Comfort	heating	free, combustion by-product	not free, heat pump or/and PTC required
	cooling	not free, AC required	not free, AC required
Components Performance, Safety and Durability	heating	simpler, components are less temperature-senstive	more complex & expensive, components highly temperature-sensitive
	cooling		
Electric resources on board	-	Not critical	Critical, decreases E-range

Chapter 2

The controlled plant

"In the 1970s, the scare was about global cooling."

— Maurice Flanagan

CONTENTS:

This chapter presents the studied controlled plant: one of the three cooling circuits in a PHEV prototype: the Golf GTE. The cooling circuit for which the NMPC will be implemented is described together with its main components.

2.1 The series vehicle

2.1.1 Hybrid Powertrain

The studied vehicle is a plug-in parallel hybrid produced by Volkswagen: the Golf GTE model shown in figure 2.1.

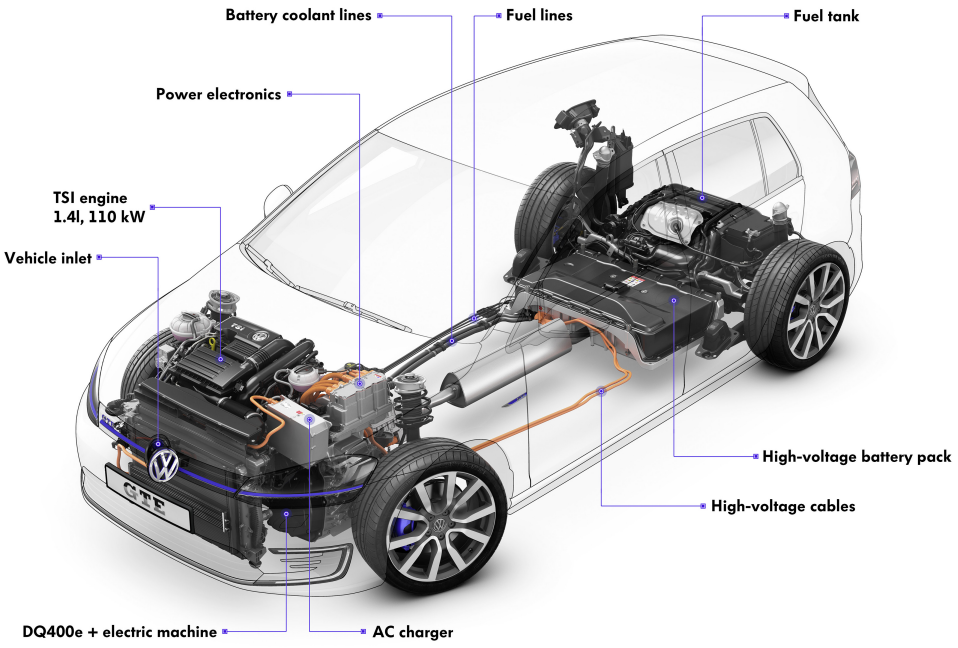


FIGURE 2.1: The studied vehicle: the PHEV Golf GTE of VW. Image taken from [49].

The main concept behind the Golf GTE is to offer a vehicle that can be driven purely electrically with an autonomy of 50km, thought for day-to-day use, or as an hybrid vehicle for longer sporadic trips. This strategy is based on the observation of many studies, such as [50], pointing that the daily average travel distance in most EU countries is around 50km as shown in figure 2.2.

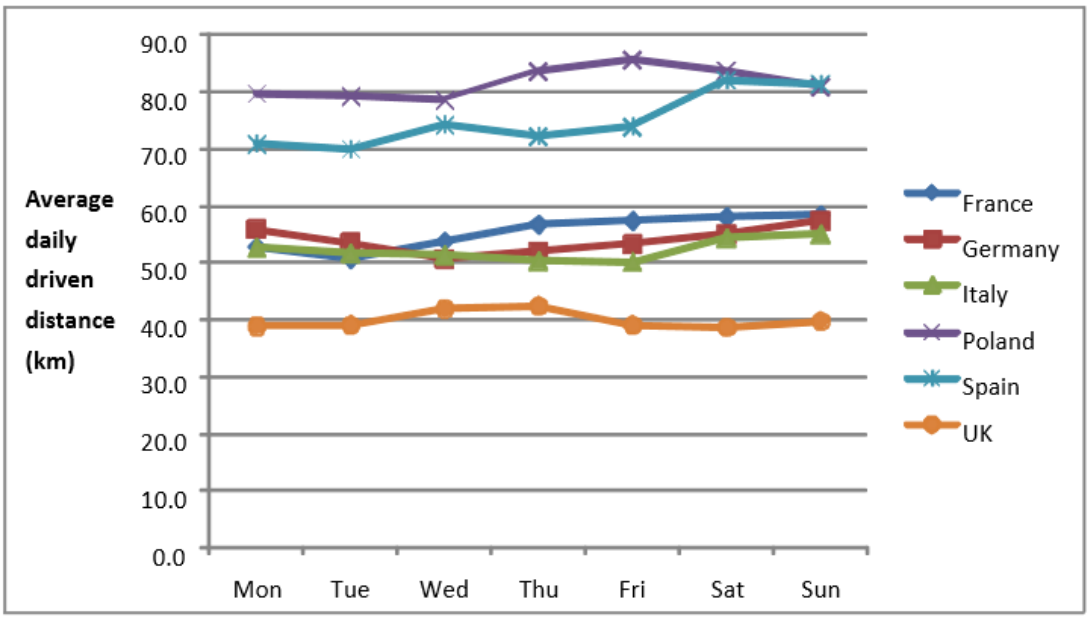


FIGURE 2.2: Average daily travel distance (km) by day of the week [50].

The powertrain of the Golf GTE can be seen in figure 2.3. It consists of the following components, which are described in detail in [51]:

- **ICE:** It is a 1.4l-TSI 110kW engine. To protect this engine during driving modes in which it is not used, design measures were employed to ensure it is always ready to deliver power immediately when needed.
- **K0:** It is a clutch used to connect/disconnect the ICE to the rest of the powertrain chain. It is designed for 350 Nm.
- **EM:** The electric machine called HEM80 is a three-phase permanently excited synchronous machine. The rotor and the stator are made of punched electric sheet steel. The rotor integrates the permanent magnets and the stator three-phase copper windings. A sensor is used to output the rotor position, a signal that is necessary for control of the phase currents through the PE.
- **K1&K2 and Gearbox:** The dual clutch gearbox DQ400E divides the flux across two component gearboxes by way of the coaxially split drive shaft with an upstream drive clutch, K1 and K2, for each. Except from the mechanically operated parking lock, control of the gears is automatic and internally handled by an electrohydraulic control module with high (40 bar) and low (5 bar) working pressure levels. The whole transmission system is highly efficient due to the low drag torques, fixed/loose bearings of the drive shafts and actuation of the drive clutches via concentric double ring pistons.

- **PE:** The PE consists of an inverter (DC/AC converter) and a DC/DC converter. The inverter incorporates six high-performance Insulated-gate Bipolar Transistors (IGBTs) (insulated-gate bipolar transistors) to convert the direct current of the high-voltage battery into a three-phase alternating current of variable frequency and voltage. This way a rotating magnetic field is created in the stator and through the attraction of the magnets field, rotor movement is created. In regenerative mode the conversion takes the other direction AC/DC. The DC/DC converter supplies the 12 V vehicle electrical system and is additionally connected to the charger. Its range is 250 to 430 V, the maximum phase current is 450 A (continuous phase current 235 A) and the switching frequency is 9kHz.
- **Battery:** The HV BAT is a modular Li-ion battery system. It is located in the vehicle's underfloor area and consists of eight modules, each made of 12 prismatic Li-ion battery cells on a nickel-magnese-cobalt basis. The energy capacity of the BAT is approximately 8.8kWh, its voltage is between 250 and 400 V depending on the state of charge.

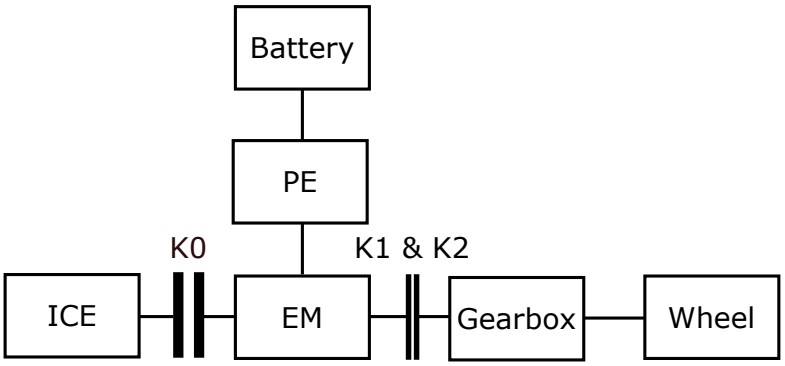


FIGURE 2.3: Golf GTE powertrain.

Depending on the combination of these components, the resulting different driving modes of Table 2.1 are possible, where SOC stands for the State of Charge of the BAT.

TABLE 2.1: Different driving modes of the Golf GTE

Mode	ICE	K0	EM	Battery	Comments
EV	off	open	Motor	Discharging	goal: 0 emissions
Hybrid Hold	on/off	closed/open	Motor/Generator	Discharging/Charging	goal: Hold SOC
Hybrid Charge	on	closed	Generator	Charging	goal: Increase SOC
Hybrid Driving	on/off	closed/open	Motor/Generator	Charging/Discharging	goal: Efficiency
Sport	on	closed	Motor	Discharging	goal: Boost

The EV mode has a maximum speed of 130 km/h and decouples the ICE to avoid its drag torque. With the different Hybrid modes both motors interact in order to achieve the different goals: in the Hybrid Hold mode the vehicle acts as a full hybrid utilising approximately 1/8 of the BAT capacity for maximum efficiency; with the Hybrid Charge mode a higher fuel consumption can be punctually paid for entering a zero-emission zone; the Hybrid Driving mode is based, among other factors, on predictive route data, in order to optimize energy consumption and finally in the Sport mode all resources are invested in achieving a powerful boost in the vehicle.

2.1.2 Cooling System for TM

A result of a complex powertrain with many different components is also a complex cooling circuit configuration for fulfilling safety and performance requirements under all possible climate conditions. In the case of the GTE, there are three different cooling circuit as shown in figure 2.4:

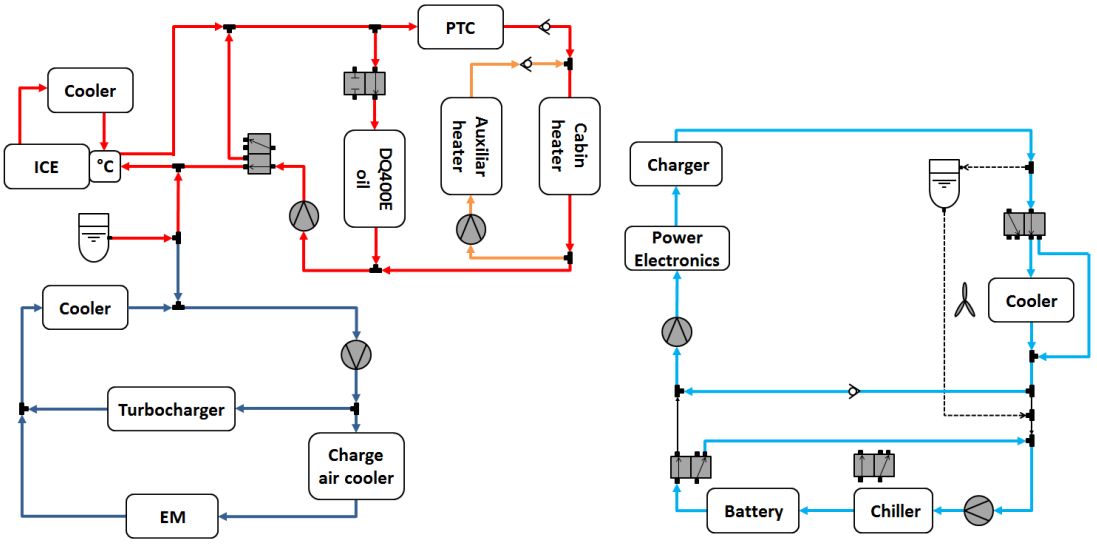


FIGURE 2.4: Golf GTE cooling circuits from [51].

The components were assigned to the following circuits according to their temperature sensitivity:

- **High-temperature (HT) circuit (red in figure 2.4):** The operating temperatures of this cooling circuit are above 90°C due to the combustion engine. The HT circuit is used to keep the engine and oil of the gearbox under optimal conditions for reducing emissions and friction losses. As in a conventional car, a thermostat and a cooler (the main water cooler) are used to identify when the engine needs to be cooled and to

cover this demand respectively. In addition, the HT circuit is responsible for the heat generation to achieve cabin comfort. This can be done by means of a heat exchanger (on the right of the HT circuit in figure 2.4) to use the heat generated in the combustion or a PTC heater used during EV mode. The optional auxiliary heater (orange in figure 2.4) is able to pre-heat the vehicle without consuming the HV BAT energy and thus avoiding a range decrease. With the electric switching valves and pumps, the volume flow through the different components can be controlled, making them act as heat sources or sinks. A compensation tank (symbol under ICE in the red circuit) is also included in this circuit to avoid over- or under pressure.

- **Low-temperature circuit 1 (dark blue in figure 2.4):** This first low-temperature circuit (LT1) conditions components with an average operating temperature of 75 to 90°C such as the charge air cooler of the ICE, the turbocharger and the EM. To satisfy this demand it uses a cooler (the Low-temperature cooler of figure 2.4). The LT1 circuit shares the compensation tank with the HT circuit.

- **Low-temperature circuit 2 (turquoise blue in figure 2.4):** The second low-temperature circuit (LT2) deals with the most temperature sensitive electric component, the BAT, the Charger and the PE. Since this circuit is the studied controlled plant, it will be described in detail in the next subsection. Here it is important to note that it includes a heat exchanger called chiller (bottom of figure 2.4). This heat exchanger enables to dissipate heat generated in the LT2 circuit in the AC system. It is located parallel to the evaporator as it can be seen in figure 2.5. Note that the purpose of this figure is only to show the LT2/AC circuit interface, hence no details of the AC architecture are given:

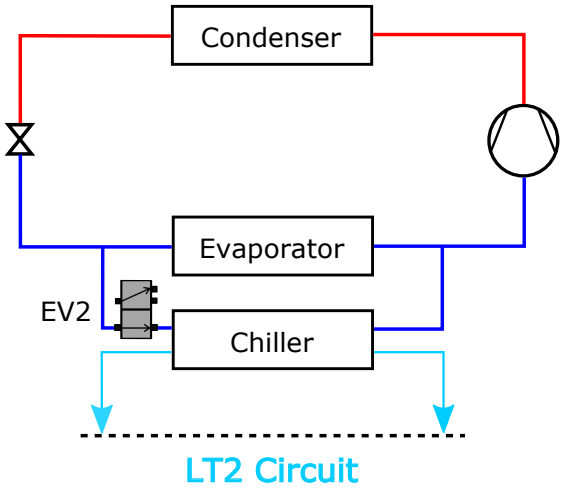


FIGURE 2.5: Golf GTE AC/LT2 circuits interface.

2.1.3 The controlled plant: the LT2 circuit

In figure 2.6 the controlled plant of this research can be seen in detail:

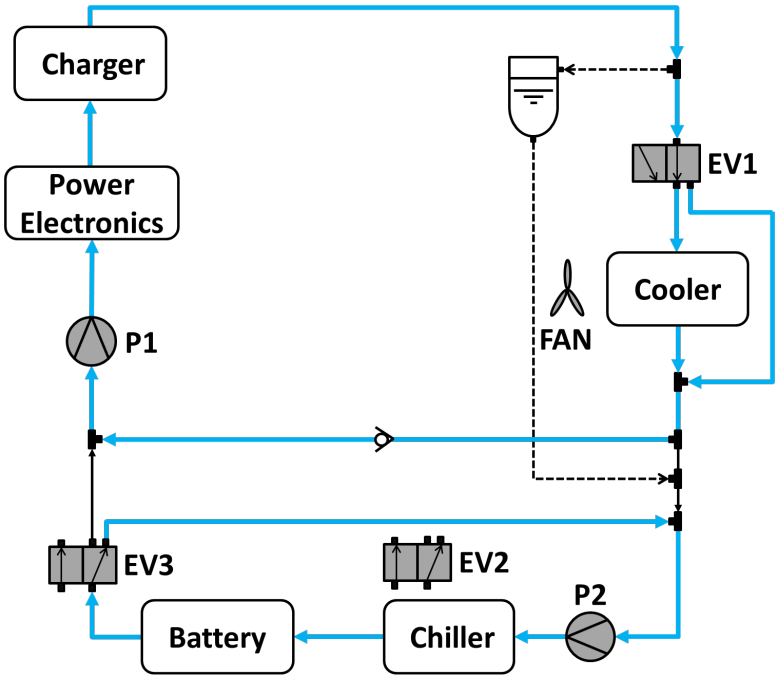


FIGURE 2.6: LT2 cooling circuit, the controlled plant.

As mentioned before, the goal of this cooling circuit is to keep the battery inside its optimal temperature range to avoid the acceleration of ageing mechanisms and commit with safety requirements. The PE and the charger need to be cooled as well. For this purpose the circuit includes two heat exchangers: the cooler and the chiller. The cooler permits the heat transfer between ambient air and coolant in front of the vehicle and the chiller, as explained before, enables the heat transfer to the AC circuit. The coolant is a water/glycol mixture and is propelled with the following electric actuators (in figure 2.6 from the top clockwise) to perform the heat transfer:

- **EV1:** this solenoid valve offers only two possible paths for the coolant: either to flow through the cooler or through its bypass. Only in the first alternative a heat transfer between air and coolant takes place.
- **FAN:** the fan is a Low Voltage (LV) consumer used to increase the air flow rate and thus improve the heat exchange between air and coolant in the cooler, it is driven by a Pulse Width Modulation (PWM) signal. It must be added that this fan is shared by the three coolers of the LT1, HT and LT2 circuits and the condenser in the AC.

- **P2:** the electrical pump P2 is also governed by a PWM signal and is used to propel the fluid providing enough pressure to overcome the pipes friction losses.
- **EV2:** this solenoid valve is the interface to the AC circuit (see figure 2.5). It has only two possible positions, either the refrigerant flows through the chiller thus accepting the LT2 rejected heat or it does not circulate through the chiller and no heat transfer is performed.
- **EV3:** also a solenoid valve, enables two paths for the coolant, thus resulting in the possible circuit configurations shown in figure 2.7. In the two circuit mode, left figure in figure 2.7, the BAT is decoupled from the PE and the charger. This way, the full cooling power of the chiller can be used to decrease the BAT temperature quickly, when needed. In the one circuit mode, right figure in figure 2.7, the BAT is coupled with the rest of the components and thus the heat transfer between all the heat sources and sinks is connected in series.
- **P1:** as the pump P1, the electrical pump P2 is also governed by a PWM signal. The need of two electrical pumps is due to the independent flow of the fluid during the two circuit mode.

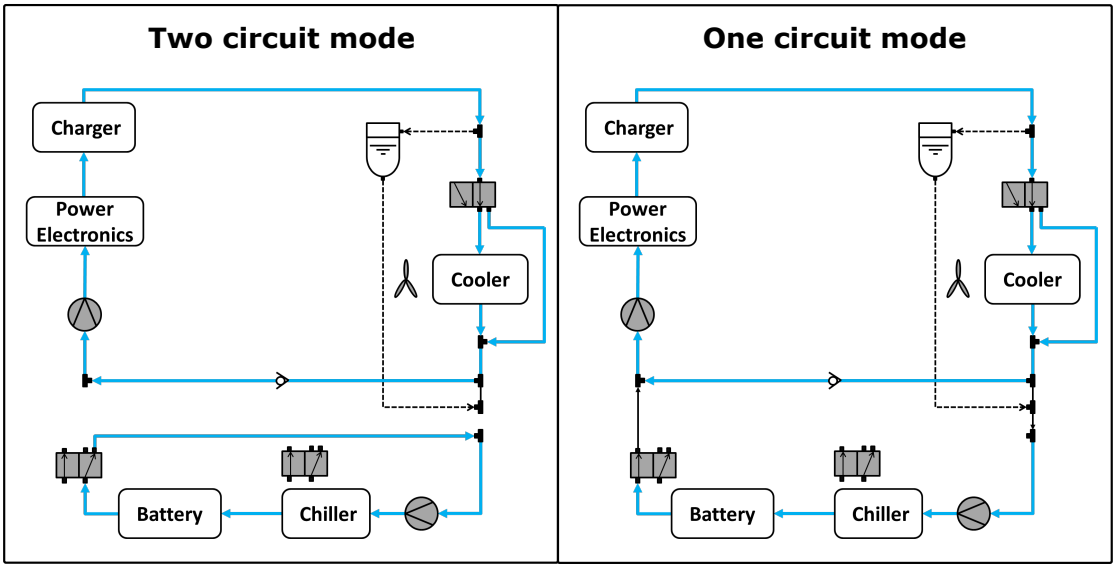


FIGURE 2.7: LT2 circuit modes.

Finally, it must be added that the LT2 circuit has its own compensation tank as shown in the top right corner of figure 2.6 and also that there are several T-junctions were the coolant is mixed. The white symbol between the two T-junctions in the middle of figure 2.6 is a non-return valve.

Chapter 3

The Model

"When I came out of school I didn't even think that modeling was a job."

— Heidi Klum

CONTENTS:

This chapter presents the core of the NMPC: the developed model of the LT2 cooling circuit. Besides an overall overview of the model, the details of each submodel will be given. All the equations presented here can also be found in the model code written in Modelica, in Appendix A.

3.1 Modeling for NMPC

3.1.1 Types of models

Modeling is a crucial step for NMPC. On the one hand, the model has to be precise enough to capture the behavior of the plant, so that the predictions done with it are sufficiently reliable. Nevertheless, on the other hand, the computational burden increases with the model complexity. Therefore, modeling for NMPC requires a trade off between accuracy and complexity. As stated in [52] there are several options for modeling:

- **First Principle Models:** are based on the analytical description of the physical understanding. A typical example is the "bicycle model" shown in figure 3.1, used for the basic vehicle dynamics, which assumes a stiff connection between two wheels which have very few degrees of freedom and one unknown coefficients: the tire-road contact.
- **Data-only Models:** which rely on a fixed-parameterized model without reference to the physics of the real plant and that needs large quantities of data to estimate its parameters.
- **Hybrid Model:** a combination of first principle and data based models.

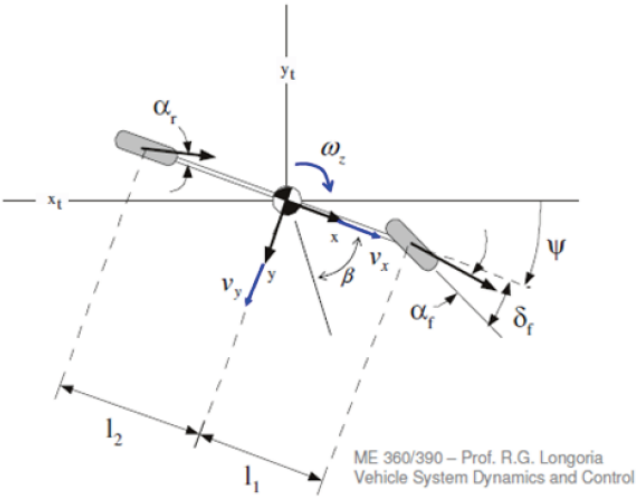


FIGURE 3.1: The bicycle model for "simple" modeling of the vehicle dynamics [53].

While first principle models suffer from their simplicity and hence difficulty to capture systems with complex transient dynamics, data-only models require a high time/resources investment for data acquisition and have to face extrapolation errors in addition. The hybrid approach combines both methods to overcome their drawbacks and is very suitable to get

simple and high performing models, exactly the NMPC requirements. For this reason, the hybrid approach was chosen for modeling in this thesis.

3.1.2 Auxiliary tools for optimization efficiency

Smooth transition function

When modeling for NMPC, there are certain requirements that need to be fulfilled to guarantee an efficient usage in the optimization process [46]. One of these requirements is determined by the numerical method behind the optimizer used in this thesis. In order to assure more efficiency in the optimization, the objective function and the constraints, thus all variables in the model, must be at least twice continuously differentiable [54]. Hence, all conditional statements have to be replaced with smooth approximations. To solve this the so called $smoothTransition$ function contained in the model library TIL was used. TIL was developed within the scope of [55] with the cooperation of TLK-Thermo GmbH and the Institute of Thermodynamics of the TU Braunschweig. From the spectrum of possible smooth transitions available in TIL, whose code can be seen in Appendix A, here the one with n = 1 was selected, so that given a transition point (x^*) and a transition length (Δx), the smooth function is calculated as follows:

$$w(x) = \begin{cases} 1 & \forall x \in (-\infty, x^* - \frac{\Delta x}{2}], \\ -\frac{1}{2} sin(\varphi) + \frac{1}{2} & \forall x \in (x^* - \frac{\Delta x}{2}, x^* + \frac{\Delta x}{2}] \\ 0 & otherwise. \end{cases} \tag{3.1}$$

where:

$$\varphi = \frac{(x - x^*)\,\pi}{\Delta x} \tag{3.2}$$

The avoidance of the discontinuities through the use of $w(x)$ instead of $f(x)$ can be seen in figure 3.2.

With the help of the smooth transition functions it can be switch a variable z between two possible states: z_0 and z_1. As an example let us consider that we have measured the electrical losses in the BAT with two experiments: once during the charging mode and once during the discharging mode thus obtaining two look-up tables with outputs z_0 and z_1. Then we can use the smooth transition function $w(x)$ to have the BAT electrical losses z described without discontinuities when changing between operation modes (charging-discharging):

$$z = w(x)z_0 + (1 - w(x))z_1 \tag{3.3}$$

where x is the BAT current and the smooth transition function $w(x)$ is defined with the transition point $x^* = 0$ and the transition length $\Delta x \longrightarrow 0$. This way, in the example, when the current sign changes, the losses calculated in z change smoothly from the output of one look-up table to the output of the other.

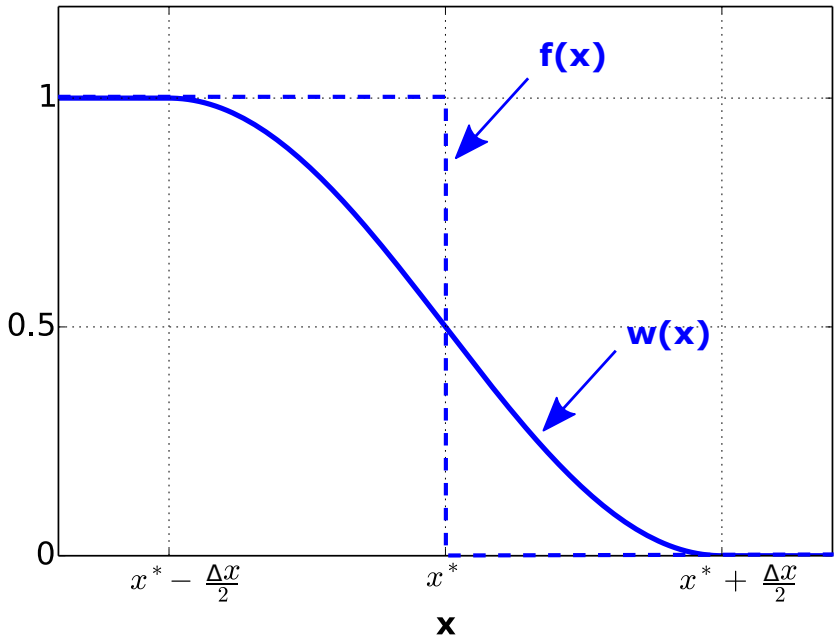


FIGURE 3.2: Discontinuities are avoided by means of the *smoothTransition* of the TIL library.

TILMedia Suite

Another crucial tool for the development of the model was the use of TILMedia Suite for Modelica [56], that provides the calculations of the thermo-physical substance properties assuming incompressibility. This way, incorporating the fluid submodels of TILMedia at every thermal port used, given the temperature, pressure and selecting the coolant, all the other properties such as enthalpy and density become effortless available.

Spline interpolation

To avoid discontinuities introduced through the usage of look-up tables, the C++ spline-interpolation-function developed in [57] and based on the methodology in [58], was used.

3.1.3 Object oriented languages

Although many NMPC applications in literature use software based on block oriented schemes in which causal relations play an important role, like Simulink or ACSL, Object Oriented (OO) acausal approaches are especially efficient in multi-domain applications [59].
Being the controlled plant in this thesis a multi-domain system where electric, thermal and hydraulic equations are needed to capture its behavior, the OO alternative was preferred. The software tool used was Dymola [60] which is based on the OO language Modelica [61].

The main difference of OO methods, compared to causal tools is that the focus of the modeling is put on the natural description of the model without consideration of the computational order. In causal tools instead, causality has to be artificially generated in order to fulfill appropriate conditions for simulation on conventional sequential computers [59]. In other words, in causal modeling tools, the engineer has to make the extra effort to describe the system with diagram block structures so that they lead to the explicit state space form:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases} \tag{3.4}$$

which can be efficiently simulated with ODEs solvers, being u the inputs, y the outputs and x the states. It is not common that a system expressed in subsystems can be naturally decomposed in such a model.
On the contrary, in Modelica it is possible to write balances and other equations as they appear in their natural form, as a system of Differential Algebraic Equations (DAEs):

$$0 = f(\dot{x}(t), x(t), y(t), u(t), t) \tag{3.5}$$

Here computer algebra is used for an automatic conversion of the equations into the ODE form and the user can concentrate only on describing the model.

Moreover, OO methods are especially suitable for modeling because they enable simple reuse of already build models thus improving error detections and reducing the model code. For an efficient modeling, the system is decomposed into subsystems (submodels) that are connected hierarchically. For example, in the LT2 cooling circuit case, the system consists of several submodels describing the pumps, the heat-exchangers, the valves and other components.

Another key aspect in OO modeling is the use of connectors, a special structure in which all the variables are collected. These variables are either of the **across**, for instance voltage in the electrical domain, or the **through** type, like current. By linking of two connectors, the submodels to which they belong are connected and matching equations for the variables are automatically generated.

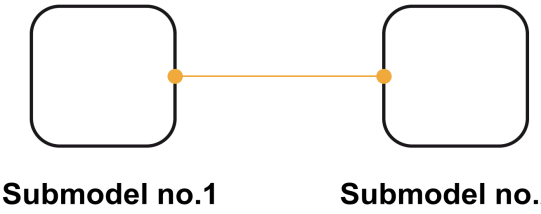


FIGURE 3.3: Connection of two submodels in Dymola.

As an example, the electrical connections of the submodels in figure 3.3 generates implicitly that the through variables sum is matched to 0 and that the across variables are equaled:

$$\begin{cases} I_1(t) + I_2(t) = 0, \\ V_1(t) = V_2(t). \end{cases} \tag{3.6}$$

In the LT2 model we find the connectors of Table 3.1, the code used in Dymola is found in the Appendix A.

TABLE 3.1: Connectors used in the LT2 model.

Connector	Cross variable	Through variable	Symbol
Thermal	$\dot{m}$	T	
LV Electric	I	U	
HV Electric	I	U	

3.2 The LT2 Model

The model of the complex nonlinear plant shown in figure 2.6 was developed in Dymola and the result can be seen in figure 3.4. In the next sections, the underlying physics of the different submodels will be described, the Modelica code of these models can be found in Appendix A.

It is useful to give a brief overview of the model's task before going into detail: The LT2 model gets as inputs the driving conditions: vehicle speed, EM torque and rotational speed,

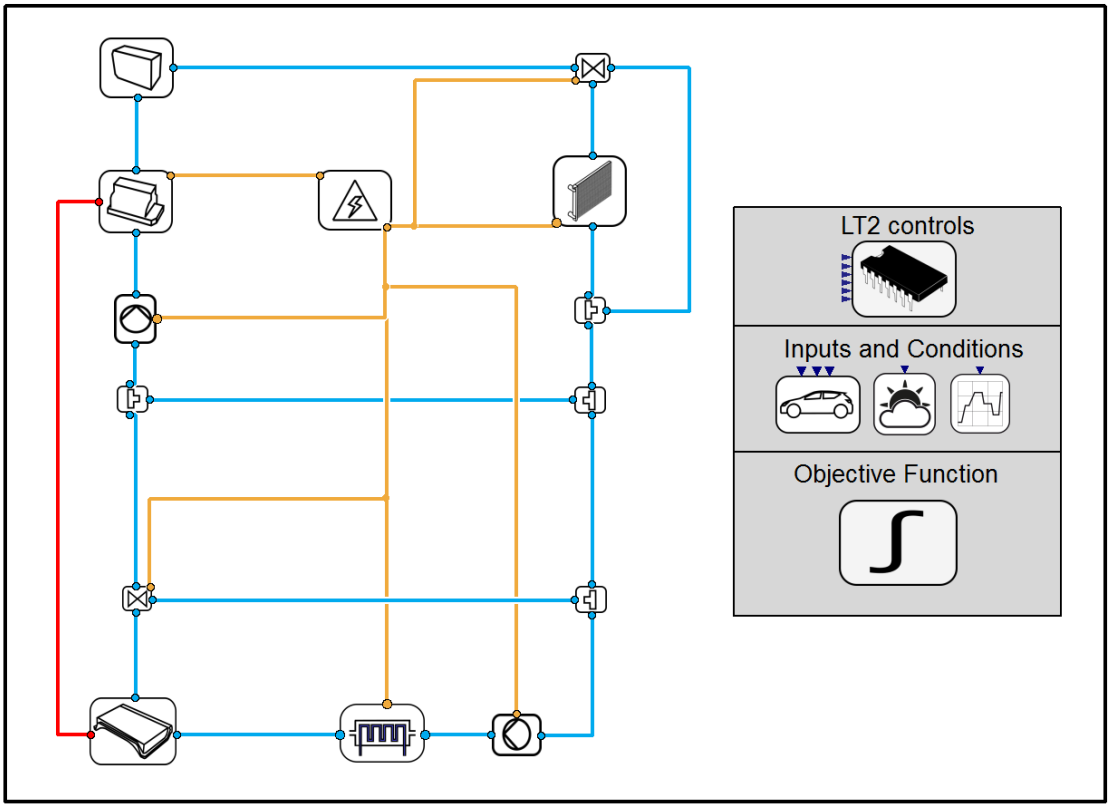


FIGURE 3.4: Developed model of the LT2 cooling circuit in Dymola.

the electrical power to move the vehicle and the ambient temperature. With these variables it is able, through electrical, thermal and hydraulic balances and several look-up tables, to determine the temperature of components and electrical consumption trajectory for the required simulation time.

In the following description of the submodels the used variables will be named. Please note that this nomenclature does not match necessarily with the one used in the Modelica code of Appendix A. This is done on purpose to facilitate the understanding of the thesis in global terms.

3.2.1 Interface submodels

These submodels are used to collect all external information and to prepare some variables needed outside the model. Hence, they are quite simple.

3.2.1.1 Car_data

The car submodel receives three inputs: the torque M and rotational speed n of the EM and the electrical traction power P_{Car} needed for moving the vehicle. For this task, it uses three Real Input connectors from the Modelica library. Calling this submodel, all the other submodels can access to its variables.

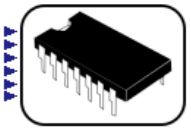
3.2.1.2 Ambient

This submodel receives the external input of the ambient temperature, $T_{ambient}$. Additionally three parameters are defined: the ambient pressure p, fixed to 1 bar, and two temperature parameters $T_{ini_{BAT}}$ and $T_{ini_{PE}}$, that can be fixed to indicate the initial temperature of the coolant in the different components depending on the circuit they belong to.

3.2.1.3 drivingCycle

This submodel receives the external input of the vehicle speed v.

3.2.1.4 Control



In this submodel the input for the six controls of the actuators are collected PWM_{fan}, PWM_{PE}, PWM_{BAT}, $Valve_{COOLER}$, $Valve_{CIRCUIT}$ and $Valve_{CHILLER}$.

3.2.1.5 Goal_function

In this submodel some variables are prepared for the later treatment in the optimization (in the goal function). First the temperature of the BAT and PE are converted to °C and the total electrical power demanded for the cooling of the LT2, P_{LT2} is calculated as:

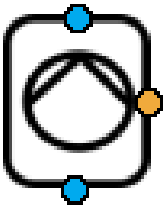
$$P_{LT2}(t) = \frac{P_{LV}(t)}{\eta_{DCDC}} + P_{Compressor}(t). \tag{3.7}$$

where P_{LV} is the power of the LV actuators calculated in the submodel lv_System, η_{DCDC} is the DC/DC converter efficiency calculated in the Powerelectronics submodel and $P_{Compressor}$ is the high voltage power demand associated to the compressor extra use needed for the heat dissipation in the chiller. This is calculated in the chiller submodel.

3.2.2 Hydraulic submodels

The hydraulic behavior of the LT2 cooling circuit is captured with the pumps, T-junctions and valves submodels. In order to reduce the number of differential states for a faster optimization, an incompressible fluid approach is taken and no pressure states are modeled. How this can be done is described within the following submodels:

3.2.2.1 Pump



The main contribution of the electric pump model is to fix the coolant volume flow rate necessary for the later heat balances. This model is based on the simulation results performed in a high fidelity model developed in the GT Suite software [62]. The high fidelity model consists of several pipes and heat exchanger elements and defines accurately the hydraulic behavior calculating the friction losses through the different geometries and materials of the elements. This approach is too detailed to be used in the NMPC model and therefore, the strategy here was to use the GT model to generate reliable look-up tables. This way, the following look-up tables were stored in each pump model:

$$\begin{aligned}
\dot{V}_{00}(t) &= f(PWM_{PE}(t), PWM_{BAT}(t), Valve_{COOLER}(t), Valve_{CIRCUIT}(t), T_{average}(t)),\\
\dot{V}_{01}(t) &= f(PWM_{PE}(t), PWM_{BAT}(t), Valve_{COOLER}(t), Valve_{CIRCUIT}(t), T_{average}(t)),\\
\dot{V}_{10}(t) &= f(PWM_{PE}(t), PWM_{BAT}(t), Valve_{COOLER}(t), Valve_{CIRCUIT}(t), T_{average}(t)),\\
\dot{V}_{11}(t) &= f(PWM_{PE}(t), PWM_{BAT}(t), Valve_{COOLER}(t), Valve_{CIRCUIT}(t), T_{average}(t)),
\end{aligned} \tag{3.8}$$

where the x, y sub-indices in $\dot{V}_{xy}$ stand for the possible positions of the circuit valve (0 two circuit mode-1 one circuit mode) and the cooler valve (0 active-1 bypass).
For each volume flow rate look-up table $\dot{V}_{xy}$ 125 simulations were run for all possible interactions between $Pwm_{PE} = [15, 30, 50, 70, 100]$, $Pwm_{BAT} = [15, 30, 50, 70, 100]$ and $T_{average} = [-10, 0, 20, 40, 60]$, thus 5 x 5 x 5 = 125 simulations. The temperature $T_{average}$ represents the average coolant temperature in the whole circuit. Finally, our Dymola model is able to select one of these four look-up tables:

$$\begin{aligned}
\dot{V}(t) = {} & (1 - Valve_{CIRCUIT}(t))(1 - Valve_{COOLER}(t))\,\dot{V}_{00}(t)\\
& + (1 - Valve_{CIRCUIT}(t))(Valve_{COOLER}(t))\,\dot{V}_{01}(t)\\
& + (Valve_{CIRCUIT}(t))(Valve_{COOLER}(t))\,\dot{V}_{11}(t)\\
& + (Valve_{CIRCUIT}(t))(1 - Valve_{COOLER}(t))\,\dot{V}_{10}(t).
\end{aligned} \tag{3.9}$$

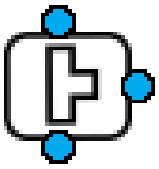
The pump model also includes a look-up table for its current demand generated from the supplier data-sheet. It depends on the PWM signal, the volume flow rate and the average temperature:

$$I(t) = f(\dot{V}(t), PWM_{PE}(t), T_{average}(t)). \tag{3.10}$$

Finally, it must be added that it is assumed that no heat transfer occurs at the pumps and that no coolant is lost or accumulated in them:

$$\begin{cases} T_{in}(t) = T_{out}(t), \\ \dot{m}_{in}(t) = -\dot{m}_{out}(t). \end{cases} \tag{3.11}$$

3.2.2.2 T-Junction



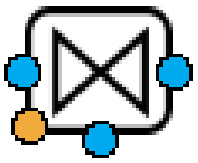
These submodels capture the behavior of the four T-Junctions along the circuit. In Appendix A they are numbered, beginning with the junction on the bottom of the cooler clockwise. Junctions 1, 3 and 4 have two coolant inlet ports and one outlet port and behind them we find a mass flow rate continuity equation and a thermal balance to reproduce the coolant mix with a certain inertia:

$$\begin{cases} \dot{m}_{in_1}(t) + \dot{m}_{in_2}(t) + \dot{m}_{out}(t) = 0, \\ \dot{m}_{in_1}(t)\, T_{in_1}(t) + \dot{m}_{in_2}(t)\, T_{in_2}(t) + \dot{m}_{out}(t)\, T_{out}(t) = 0.1\, \frac{dT_{out}(t)}{dt}. \end{cases} \tag{3.12}$$

Junction 2 has one inlet and two outlets, it splits the flow rate coming from the cooler to the PE circuit or to the BAT circuit. Following simple balances are used to capture its dynamics:

$$\begin{cases} \dot{m}_{out_1}(t) + \dot{m}_{out_2}(t) + \dot{m}_{in}(t) = 0, \\ T_{out_1}(t) = T_{out_2}(t) = T_{in}(t). \end{cases} \tag{3.13}$$

3.2.2.3 Solenoid cooler and circuit valves



The solenoid valves let the inlet coolant flow through one of their two outlet ports depending on the control signal. In the case of the circuit valve this is achieved with the following equations:

$$\begin{cases} \dot{m}_{out_1}(t) = -(1 - Valve_{CIRCUIT}(t))\,\dot{m}_{in}(t), \\ \dot{m}_{out_2}(t) = -Valve_{CIRCUIT}(t)\,\dot{m}_{in}(t). \end{cases} \tag{3.14}$$

On the contrary, the cooler valve determines the volume flow rate through the cooler with the help of look-up tables obtained in the same high fidelity model simulations as the pumps. Since, as mentioned before, no pressure states are modeled, the compensation tank behavior is represented only through its influence on the amount of coolant volume flow rate that passes through the cooler. This way, analog to the pump models, the mass flow rate through the cooler is calculated as follows:

$$\begin{aligned} \dot{V}_{out1}(t) = {} & (1 - Valve_{CIRCUIT}(t))(1 - Valve_{COOLER}(t))\,\dot{V}_{00}(t) \\ & + (1 - Valve_{CIRCUIT}(t))(Valve_{COOLER}(t))\,\dot{V}_{01}(t) \\ & + (Valve_{CIRCUIT}(t))(Valve_{COOLER}(t))\,\dot{V}_{11}(t) \\ & + (Valve_{CIRCUIT}(t))(1 - Valve_{COOLER}(t))\,\dot{V}_{10}(t). \end{aligned} \tag{3.15}$$

$$\dot{V}_{out1}(t) = -\dot{m}_{out}(t)\,\frac{60000}{\rho(t)}. \tag{3.16}$$

where ρ is the density calculated by means of TIL Media. And then to obtain the coolant skipping the cooler, an easy mass flow balance is done:

$$\dot{m}_{out2}(t) + \dot{m}_{in}(t) + \dot{m}_{out1}(t) = 0. \tag{3.17}$$

Finally, notice that in the solenoid valve, no heat is transfered to the coolant

$$T_{out2}(t) = T_{in}(t) = T_{out1}(t) \tag{3.18}$$

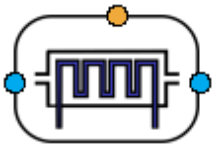
and that only in one position, the valves consume 15.4W. This position is determined for each valve according to a safety requirement, so that in case of supply failure, the valve remains in the position that enables the cooling of the components. For this reason, the cooler valve consumes at the bypass and the circuit valve in the one circuit mode:

$$\begin{cases} I_{valve}(t) = \frac{15.4\,Valve_{CIRCUIT}(t)}{U_{LV}}, \\ I_{valve}(t) = \frac{15.4\,Valve_{COOLER}(t)}{U_{LV}}. \end{cases} \tag{3.19}$$

3.2.3 Thermal submodels

In this section we present the submodels of the two heat sinks: the cooler and the chiller.

3.2.3.1 Chiller



As explained before, the chiller is a heat exchanger parallel to the evaporator in the AC circuit. When active, the chiller can dissipate up to 1350W. The heat transfer is not an automatic response to the valve command, therefore this was modeled with a first order system with time constant τ:

$$\frac{d\dot{Q}_{CHILLER}(t)}{dt} = \frac{-1350\,Valve_{CHILLER}(t) - \dot{Q}_{CHILLER}(t)}{\tau}. \tag{3.20}$$

Depending on the heat dissipated in the AC circuit and the Coefficient Of Performance (COP), the extra power required in the compressor for the LT2 task, can be calculated as follows:

$$\begin{cases} P_{Compressor}(t) = \frac{-\dot{Q}_{CHILLER}(t)}{COP(t)}, \\ COP = f(\dot{m}_{air}(t)). \end{cases} \tag{3.21}$$

where the COP depends, in turn, on the mass flow rate in front of the vehicle where the condenser and the coolers are placed. The LV consumption of the solenoid valve in the chiller responsible for its activation can be calculated as in the case of the other two solenoid valves:

$$I_{valve}(t) = \frac{15.4(1 - Valve_{CHILLER}(t))}{U_{LV}}. \tag{3.22}$$

Once the chiller heat rejection is known, the first law of thermodynamics can be applied to calculate the different temperatures:

$$\frac{dU_i}{dt} = \dot{Q}_{thm}(t) = \dot{Q}_{CHILLER}(t) - \dot{Q}_{ambient}(t) - \dot{Q}_{coolant}(t), \tag{3.23}$$

where

$$\dot{Q}_{thm}(t) = m\, c_{component}\, \frac{dT_{component}(t)}{dt}, \tag{3.24a}$$

$$\dot{Q}_{convection}(t) = A\,\alpha(t)\,(T_{component}(t) - T_{ambient}(t)) \tag{3.24b}$$

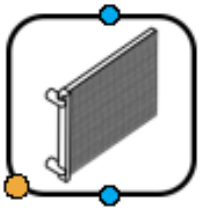
$$\dot{Q}_{radiation}(t) = A\,\epsilon\,\sigma\,(T^4_{component}(t) - T^4_{ambient}(t)), \tag{3.24c}$$

$$\dot{Q}_{ambient}(t) = \dot{Q}_{convection}(t) + \dot{Q}_{radiation}(t), \tag{3.24d}$$

$$\dot{Q}_{coolant}(t) = \dot{m}(t)\, c_p(t)\,(T_{outlet}(t) - T_{inlet}(t)), \tag{3.24e}$$

being U_i, m, c, A, ϵ the internal energy of the system, the chiller mass, specific heat capacity, surface and emissivity, respectively. $\sigma = 5.67\,10^8$ is the Boltzmann constant and α the convection coefficient for the component-air heat transfer in the submodel Coefficient calculations. The emissivity ϵ was taken constant 0.8 and finally the heat capacity $m\,c$ and A of the component were calibrated with experimental measurements. The fluid heat capacity c_p and other properties such as the density are calculated as usual by means of TILMedia.
Notice that it is assumed that the surface in contact between the refrigeration pipes and the heat source in the component is enough large to ensure a fast equilibrium and thus $T_{outlet} = T_{component}$.

3.2.3.2 Cooler



The dissipated heat flow through the cooler is determined with the Number of Transfer Units (NTU) method:

$$\dot{Q}_{air}(t) = h(\dot{V}(t), \dot{m}_{air}(t))\,(T_{in}(t) - T_{ambient}(t)). \tag{3.25}$$

where the heat capacity rate h was observed experimentally and stored in a look-up table with the air mass and the coolant volume flow rate as inputs. The inlet temperature of the air was assumed to be the ambient and its mass flow was determined with a polynomial fruit of experience that depends on the PWM_{FAN} signal and the vehicle speed:

$$\dot{m}_{air}(t) = f(v(t), PWM_{fan}(t)). \tag{3.26}$$

Again a thermal balance as in equation 3.23 is done for calculating the temperatures:

$$\frac{dU_i(t)}{dt} = \dot{Q}_{thm}(t) = -\dot{Q}_{air}(t) - \dot{Q}_{ambient}(t) - \dot{Q}_{coolant}(t). \tag{3.27}$$

Finally, the LV electrical current demanded for moving the fan comes from a polynomial obtained from measurements:

$$I_{LV}(t) = f(PWM_{fan})(t). \tag{3.28}$$

3.2.3.3 Coefficient calculations

In this submodel, the convection coefficient α is calculated with the empirical Nusselt correlations for forced air convection in flat plates [63] with a characteristic length $l_m = 1.5m$:

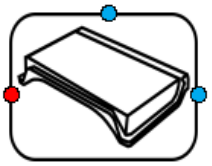
$$\begin{cases} \overline{T}(t) = \frac{T_{component}(t)+T_{ambient}(t)}{2}, \\ Re = \frac{(v_{air}(t)+1)l_m\dot{\rho}(\overline{T}(t))}{\vartheta(\overline{T}(t))}, \\ Nu_{lam}(t) = 0.664\sqrt{Re(t)}\,Pr(\overline{T}(t))^{\frac{1}{3}}, \\ Nu_{turb}(t) = \frac{0.037\,Re(t)^{0.8}\,Pr(\overline{T}(t))}{1+2.443\,Re(t)^{-0.1}\,(Pr(\overline{T}(t))^{\frac{2}{3}}-1)}, \\ Nu(t) = 0.2 + \sqrt{Nu_{lam}^2(t) + Nu_{turb}^2(t)}, \\ \alpha(t) = \frac{Nu(t)\,\lambda(\overline{T}(t))}{l_m}, \end{cases} \tag{3.29}$$

where $Re(t)$ is the Reynolds number, $Nu_{lam}(t)$ and $Nu_{turb}(t)$ the Nusselt correlation for the laminar and turbulent case, $Pr(t)$ the Prandtl number and $\vartheta(t)$ and $\lambda(t)$ the kinematic viscosity and the thermal conductivity, respectively.

3.2.4 Electrical submodels

In this section we present the electric components that have to be cooled with the LT2 cooling circuit. Although they are organized in this subsection, they include balances in different domains, as it will be shown as follows:

3.2.4.1 HV Battery



The HV Battery is the most important component in the model. The voltage in the battery is the open-circuit voltage measured depending on the State Of Charge (SOC) and the temperature and is stored in look-up tables:

$$U_{HV}(t) = f(SOC(t), T_{component}(t)). \tag{3.30}$$

The SOC is also a polynomial dependent on the BAT energy:

$$SOC(t) = f(E_{BAT}(t)). \tag{3.31}$$

The energy in the BAT is calculated as follows:

$$\frac{dE_{BAT}(t)}{dt} = P_{HV}(t) - P_{loss_{BAT}}(t). \tag{3.32}$$

where the high voltage electric power is $P_{HV}(t) = U_{HV}(t)I_{HV}(t)$ and the losses due to the joule effect $P_{loss_{BAT}}(t) = Ri(t)I_{HV}^2(t)$. The internal resistance of the battery $Ri(t)$ is obtained from the combination of two look-up tables with measurements in the charging and discharging case by the smooth transition factor $wf_{Ri(t)}$:

$$\begin{cases} Ri_{charge}(t) = f(SOC(t), T_{component}(t)), \\ Ri_{discharge}(t) = f(SOC(t), T_{component}(t)), \\ Ri(t) = wf_{Ri}(t)Ri_{discharge}(t) + (1 - wf_{Ri}(t))Ri_{charge}(t), \\ wf_{Ri}(t) = smoothTransition(P_{HV}(t), 0, 0.1). \end{cases} \tag{3.33}$$

Once the losses in the BAT are known, its effect on the coolant and component temperature can be derived from the same heat balance as in the other thermal components:

$$\frac{dU_i(t)}{dt} = \dot{Q}_{thm}(t) = P_{loss_{BAT}}(t) - \dot{Q}_{ambient}(t) - \dot{Q}_{coolant}(t). \tag{3.34}$$

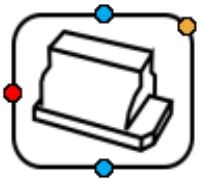
3.2.4.2 Low voltage net

In the LV net, the different current demands in the actuators are collected and summed with the average of all other LV consumers in the vehicle given by $I_{auxiliary} \approx 15A$. This assumption is based on several real cycles available data with lights, radio and AC off:

$$\begin{aligned} I_{LV}(t) = I_{PWM_{BAT}}(t) + I_{PWM_{PE}}(t) + I_{PWM_{fan}}(t) + I_{Valve_{CHILLER}}(t) \\ + I_{Valve_{COOLER}}(t) + I_{Valve_{CIRCUIT}}(t) + I_{auxiliary} \end{aligned} \tag{3.35}$$

An thus the total LV power demand can be calculated as $P_{LV}(t) = I_{LV}(t)U_{LV}$, where U_{LV} is assumed to be constant and equal to 14 V.

3.2.4.3 Power Electronics



In the PE the two electric levels, LV and HV, are managed. First the LV required to satisfy the consumers demand and the DC/DC losses are calculated assuming a constant efficiency of $\eta_{DCDC} = 90\%$ in the converter:

$$\begin{cases} P_{loss_{PE_{LV}}}(t) = P_{LV}(t)\,\frac{1-\eta_{DCDC}}{\eta_{DCDC}}, \\ P_{LV_{required}}(t) = P_{loss_{PE_{LV}}}(t) + P_{LV}(t). \end{cases} \tag{3.36}$$

And the high voltage required have to be enough to overcome the traction of the vehicle, $P_{car}(t)$, to feed the compressor if the chiller needs it, $P_{Compressor}(t)$, and finally to overcome the inverter losses $P_{loss_{PE_{HV}}}(t)$:

$$P_{HV_{required}}(t) = P_{Car}(t) + P_{Compressor}(t) + P_{loss_{PE_{HV}}}(t). \tag{3.37}$$

For calculating the losses in the DC/AC inverter again look-up tables, generated with measurements depending on the torque and rotational speed of the electric machine and the BAT voltage, were used:

$$\begin{cases} P_{loss_{motor}}(t) = f(M(t), n(t), U_{HV}(t)), \\ P_{loss_{generator}}(t) = f(M(t), n(t), U_{HV}(t)), \\ P_{loss_{PE_{HV}}}(t) = -wf_{Ploss}(t)\,P_{loss_{generator}}(t) - (1 - wf_{Ploss}(t))\,P_{loss_{motor}}(t), \\ wf_{Ploss}(t) = smoothTransition(M(t), 0, 0.1). \end{cases} \tag{3.38}$$

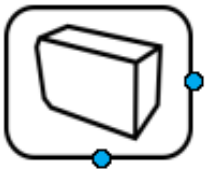
With the LV and HV requirements it calculates the total energy to be supplied by the battery is calculated as:

$$P_{HV}(t) = U_{HV}(t)\,I_{HV}(t) = P_{HV_{required}}(t) + P_{LHV_{required}}(t). \tag{3.39}$$

And the total losses in the PE, $P_{loss_{PE}}(t) = P_{loss_{PE_{HV}}}(t) + P_{loss_{PE_{LV}}}(t)$, are needed for its thermal balance:

$$\frac{dU_i(t)}{dt} = \dot{Q}_{thm}(t) = P_{loss_{PE}}(t) - \dot{Q}_{ambient}(t) - \dot{Q}_{coolant}(t). \tag{3.40}$$

3.2.4.4 Charger



As explained before, since the charging state is not studied in this research, the role of the charger in the LT2 is just a heat sink with:

$$\frac{dU_i(t)}{dt} = \dot{Q}_{thm}(t) = -\dot{Q}_{ambient}(t) - \dot{Q}_{coolant}(t). \tag{3.41}$$

3.3 Model validation

To validate the model, six different driving cycles were performed in the road with a PHEV prototype as it can be seen in figure 3.5. All of them draw from the SEAT Technical Centre, located in Martorell, a city 32 km from Barcelona and 10 km from the Montserrat mountain and were chosen with the aim of forming an heterogeneous set. Therefore they present different average speeds, as it can be seen in the third column in Table 3.2 and they can be classified in several categories, second column in the mentioned table, with the help of the nomenclature defined in [64]. Furthermore, the cycles were performed under different ambient temperatures, as shown in the fourth column in Table 3.2

During each driving cycle the controls, inputs and states of the LT2 model were logged with the data acquisition system that will be explained later, in chapter 6. After driving, the measured inputs and controls were used at the desk to simulate the model. The resulting simulation plant states were then compared to the values measured in the real vehicle where the focus was put in the fitting of the BAT and PE temperature trajectories. The error ε between the simulation results and the measures is defined as:

$$\begin{aligned} \varepsilon_{BAT}(t) &= T_{component_{BAT}}(t) - T_{BAT_{Real}}(t) \\ \varepsilon_{PE}(t) &= T_{component_{PE}}(t) - T_{PE_{Real}}(t) \end{aligned} \tag{3.42}$$

The last four columns in Table 3.2 show for each driving cycle, the mean μ and standard deviation σ of these errors.

It must be highlighted that the BAT average error and standard deviation never exceed 1°C, while the PE presents an average error always below 2°C with a deviation below 3°C. As stated before, the goal of the model in a predictive control is to grasp the behavior being, at the same time, fast enough for computation.

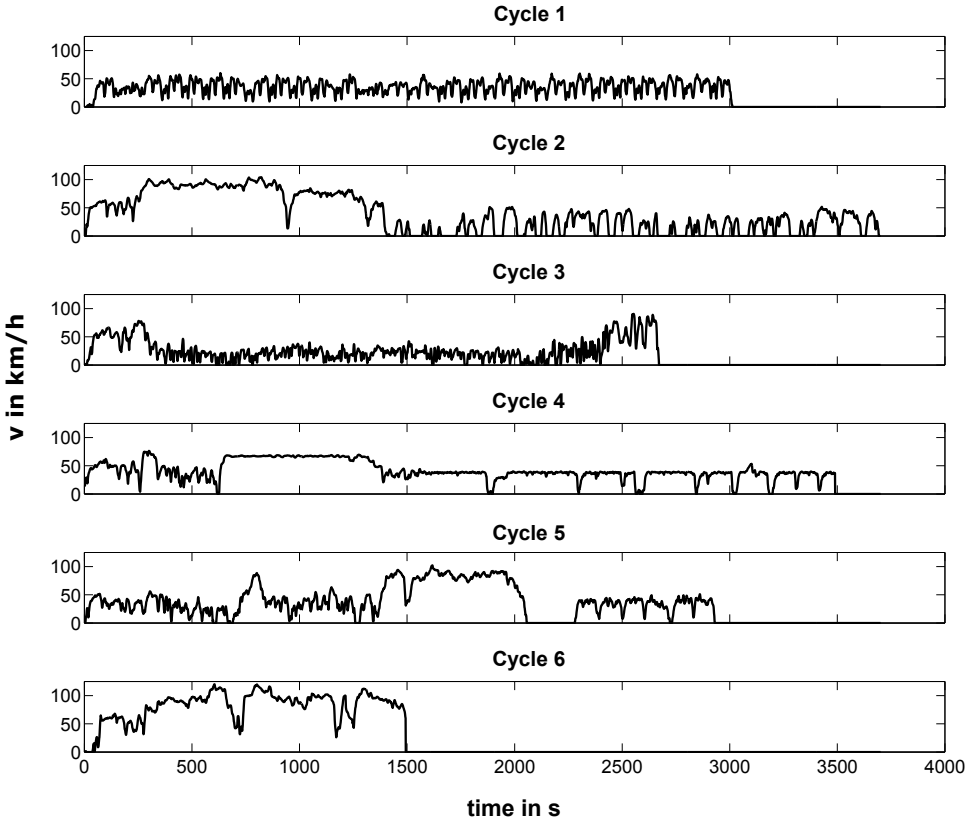


FIGURE 3.5: Driving cycles set chosen for model validation.

Among the driving cycles, Cycle 4 in figure 3.5 draws the attention due to its flat segments in the speed profile. The reason for this special profile will be revealed in chapter 6 since this is one of the two driving cycles used later for the NMPC validation. Hence, given its importance, it is worth to take a closer look at its model predicted and measured temperature trajectories in figure 3.6. The figure shows the goodness of the model for the chosen driving cycle, since it can be clearly seen that the model captures the transient dynamics of the components temperatures reliably: the PE and BAT predicted trajectories (red and blue lines) follow the tendency of the measurements (black lines).

TABLE 3.2: Validation results for different cycles.

Name	Type/Road	$\mu_{speed}(km/h)$	$\mu_T(^\circ C)$	$\mu_{\varepsilon_{PE}}(^\circ C)$	$\sigma_{\varepsilon_{PE}}(^\circ C)$	$\mu_{\varepsilon_{BAT}}(^\circ C)$	$\sigma_{\varepsilon_{BAT}}(^\circ C)$
Cycle 1	Urban, free-flowing	35,5	16,7	1,0	0,8	-0,2	0,6
Cycle 2	Motorway + urban unsteady*	39,4	15,0	0,8	-2,4	-0,9	0,5
Cycle 3	Urban, unsteady	26,8	24,1	1,2	0,0	-0,5	0,5
Cycle 4	Rural road + slope	41,8	22,5	1,1	-1,1	-0,8	0,5
Cycle 5	Motorway + secondary rural roads	41,1	20,4	1,2	-2,8	-0,4	0,3
Cycle 6	Main road, steady speed	80,9	22,8	1,8	0,0	0,0	0,7

*cycle with more frequent acceleration and deceleration

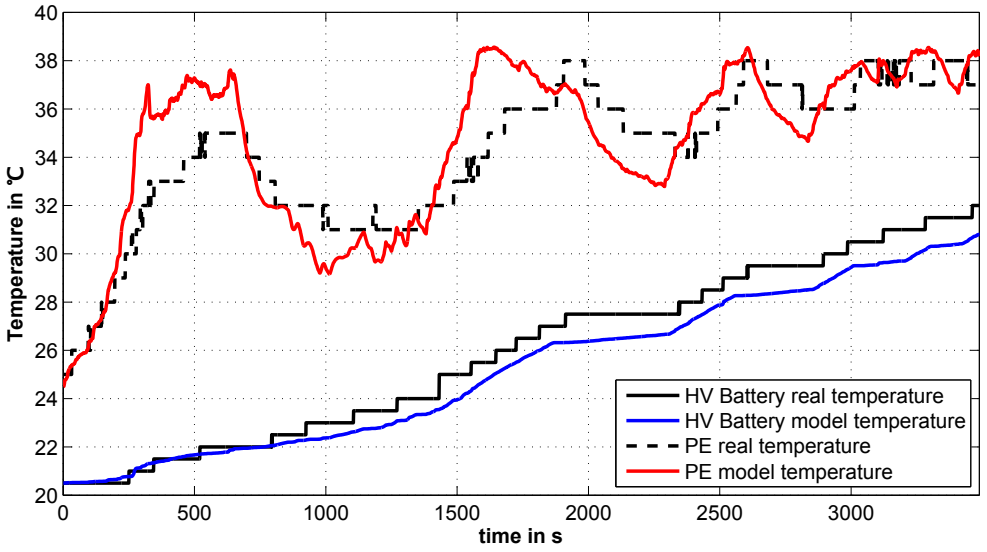


FIGURE 3.6: Model validation for the Cycle4.

On the other hand, it must be said that to validate the simplicity of the model required for a low computational burden in the optimization process, three different cycles [65] were used to measure the real time needed for their simulation. The duration of the cycle and the average time of five simulations using the DASSL solver of Dymola can be seen in the second and third columns of Table 3.3.

TABLE 3.3: Real time required for several cycles simulations.

Cycle	Duration(s)	Simulation(s)	Factor(-)
EMPA B	2024	4.89	>400
EMPA Bschl	963	2.97	>300
EMPA BAB	1000	1.05	>900

Taking a look at the last column of Table 3.3, it can be concluded that despite its complexity (around 500 equations, 1300 variables, 6 controls and 15 differentiated state variables), the model it is still hundreds of times faster than real time.

As mentioned before, when modeling for NMPC the simplicity/accuracy trade-off is crucial. In this sense, it can be concluded that the build model is suitable for NMPC since, as shown in this chapter, it captures the dynamics of the system with a limited error and this is achieved despite the simple modeling approach taken which uses only one single temperature for each component in the circuit, among other simplifications. Moreover, Table 3.3 suggests that the model is suitable for online optimization, although this will be validated during the optimization preparation process of chapter 5.

Chapter 4

Optimal Control Problem formulation

> "A problem is a chance for you to do your best."
>
> — Duke Ellington

CONTENTS:

The problem of controlling a dynamic system opens the door to different strategies. In this thesis we use optimization methods, as suggested in the quotation, to play the right note at the right time and achieve the best possible performance of the system. This chapter introduces the optimization with the OCP formulation, reviews briefly the most popular approaches for solving OCPs and finally gives more details about the special numerical methods behind MUSCOD-II.

4.1 Optimal Control Formulation: general description

An optimization problem is the mathematical task of finding the values of the variables $\xi \epsilon \mathbb{R}^n$ that minimize a certain function $\Phi(\xi) : \mathbb{R}^n \longrightarrow \mathbb{R}$:

$$\min_{\xi} \quad \Phi(\xi) \tag{4.1}$$

$\Phi(\xi)$ describes some objective, a quantitative measure of the performance of the system under study that we want to minimize and hence it is usually called cost, goal or objective function. Furthermore, it is usual that the variables of the system that can minimize such function are constrained or restricted, like for example the mass of a body can take only positive values. In this case, the goal is to find the variables from the feasible region, the set of points satisfying all the constraints, that minimize the objective function:

$$\begin{aligned} \min_{\xi} \quad & \Phi(\xi) \\ \text{subject to} \quad & c(\xi) \leqslant 0 \end{aligned} \tag{4.2}$$

where $c(\xi) : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ stands for p constraints. It must be added that $c(\xi)$ and $\Phi(\xi)$ must be C^2. The reason for this condition is that the smoothness of the functions makes it possible to use their information at a particular point x to deduce information about the function's behavior at all points close to x and this feature is necessary for the algorithms to move along the function in the minimum search.

Optimization problems that can be directly solved with numerical methods are called programs. If both the cost function and the constraints are linear functions of ξ, we are faced with a Linear Program (LP), otherwise with a Nonlinear Program (NLP). Depending on the particular form or structure of the functions, they can be further divided in other categories such as Quadratic Program (QP). For a deeper insight in the numerical solution of the different types of programs, the reader is referred to [66].

So far, the cost function and the described constraints are static functions independent from the variable time. Nevertheless, in the control theory field, an interdisciplinary branch of engineering and mathematics, the optimization problems are focused on physical systems that can, in some cases, be described with the help of ODEs where the independent variable is the time t:

$$\dot{x}(t) = \frac{dx}{dt}(t) = f(x(t), u(t), d, p). \tag{4.3}$$

Depending on the current state of the system $x(t) \in \mathbb{R}^{n_x}$, the control inputs $u(t) \in \mathbb{R}^{n_u}$, the time-invariant parameters $p \in \mathbb{R}^{n_p}$ and some uncontrolled inputs called disturbances d, the ODEs system describes the evolution of the states over time. The usual objective of the control strategy is to drive the system to a convenient or desired output. The approach of formulating the control problem with an optimization problem structure, as shown before, and using numerical methods to find the best possible controls is referred as Optimal Control Problem (OCP). In this research we are interested in OCP of the form:

$$\min_{x(\cdot),u(\cdot)} \quad \Phi(x(\cdot), u(\cdot), d, p) \tag{4.4a}$$

$$\text{s.t.} \quad 0 = x(t_0) - \overline{x}_0, \tag{4.4b}$$

$$\dot{x}(t) = f(x(t), u(t), d, p) \qquad \forall t \epsilon H_p, \tag{4.4c}$$

$$0 \leqslant c(x(t), u(t), d, p) \qquad \forall t \epsilon H_p. \tag{4.4d}$$

The goal here is to find the control $u(\cdot)$ and consequently state trajectories $x(\cdot)$ that minimize the objective function Φ over an optimization horizon $H_p := [t_0, t_f]$. At the same time, the solution must satisfy the relations described in the ODEs system (4.4c), some path constraints $c(x(t), u(t), d)$ (4.4d) and must draw from the initial values of the states $\overline{x}_0$ (4.4b).

4.2 Optimal Control Formulation: LT2 cooling circuit

The formulated OCP for the control of the LT2 circuit will be described in the following subsections. In the optimization tool used here, MUSCOD-II, the OCP formulation is introduced by means of two separate files: the so called .DAT file, which is written in ASCII and uses the keywords defined in [48] and the .cpp file which is in C++. With the first one, the user provides information about the states and controls constraints, the initialization and scaling of the problem and can change some optimizer settings. In the later, the model and the objective function are given to the optimizer and additionally path or point constraints can be defined. In Appendix B an example of these files is included to facilitate the understanding of the optimizer setting process described in this section.

4.2.1 Objective function

As already introduced, the goal of the control of the LT2 cooling circuit is to achieve an accurate and efficient TM. To evaluate the accuracy and efficiency of the TM, two cost terms were defined: c_T and c_P.

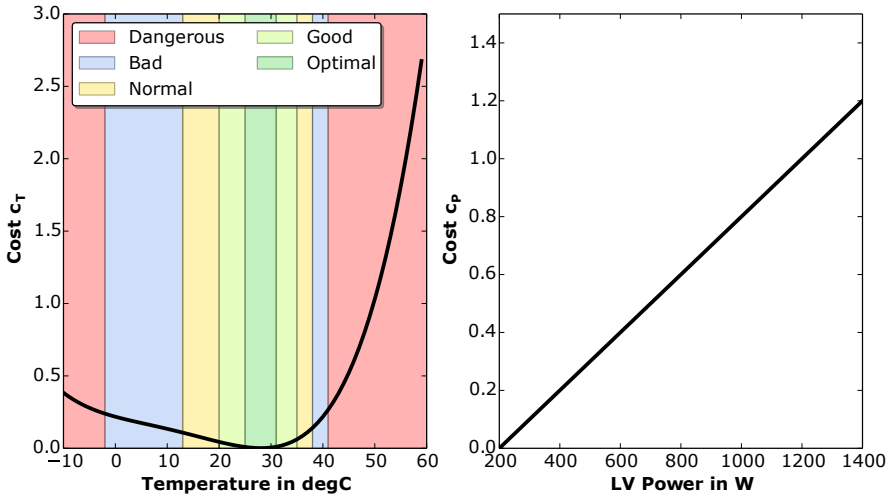


FIGURE 4.1: Cost terms included in the objective function to evaluate the accuracy and efficiency of the TM.

The cost term c_T (in the left of figure 4.1) describes with the following polynomial the effect of the working temperature on the battery, so that the further from the optimal range, the more promoted the aging mechanisms and/or the higher the BAT performance drop is:

$$c_T(T) = a_4\,T^4 - a_3\,T^3 + a_2\,T^2 - a_1\,T + a_0. \tag{4.5}$$

where $a_0, a_1 ... a_4$ are constant factors and $T = T_{outlet_{BAT}} = T_{component_{BAT}}$.
The cost term c_P (on the right of figure 4.1) is the following linear function depending on the electrical power of the actuators P_{LT2}:

$$c_P(P_{LT2}) = \frac{P_{LT2} - b_0}{b_1}. \tag{4.6}$$

where again b_0, b_1 are constant factors. c_P indicates that the more electrical power it is used for the TM, the less attractive it is. Table 4.1 shows the used electrical actuators categorizing them according to the amount of electric power they require.

TABLE 4.1: Actuators electrical power.

Actuator	Control Signal	Electrical power
Cooler valve	$Valve_{COOLER} \in \{0, 1\}$	low
Fan	$PWM_{FAN} \in [10, 90]$	high
BAT pump	$PWM_{BAT} \in [0, 100]$	medium
Chiller valve	$Valve_{CHILLER} \in \{0, 1\}$	low
Compressor	$Valve_{CHILLER} \in \{0, 1\}$	high
Circuit valve	$Valve_{CIRCUIT} \in \{0, 1\}$	low
PE pump	$PWM_{PE} \in [0, 100]$	medium

The total cost associated to the TM is given by c, which is the sum of the two penalty terms of figure 4.1 as it can be seen in the following expression:

$$c = c_T + c_P. \tag{4.7}$$

In the line 174 of the .cpp file included in Appendix B the objective function is defined as a Lagrange-type of the following form:

$$\Phi = \int_{t_0}^{t_f} c \, dt \tag{4.8}$$

As it can be seen in the lines 145-146 of the .cpp file and lines 221-223 of the .dat file, the total cost function can be tuned with the following factors ($coef_T$, $coef_P$ and $time_{mean}$):

$$c = \frac{coef_T \; c_T + coef_P \; c_P}{time_{mean}}. \tag{4.9}$$

The factors $coef_T$ and $coef_P$ are used to adjust or calibrate the relative importance of the two objectives while $time_{mean}$ is the total duration, exact or approximated if unknown, of the driving cycle that will be optimized. The latter is used to scale the objective function with tangible orders of magnitude around the costs terms of figure 4.1.

4.2.2 Constraints

Equation (4.4c) implies the writing down in C++ of the 500 equations in the LT2 model resulting from the combination of the explicitly written submodels and the automatic generated connectors equations. Since this is complicate and can easily lead to transciption errors, the methodology proposed in [67] was taken.

The main idea is to benefit from the standardized model exchange format Functional Mock-up Interface (FMI) [68, 69] exporting the model in Dymola as an Functional Mock-up Unit (FMU) and importing it to MUSCOD-II with the software framework of the mentioned article. The FMU function calls for reading the model information can be seen in the .cpp file in Appendix B. Moreover, in this research we used a Graphical Users Interface (GUI) developed by Dr. Gräber, author of [67], that with the FMU as input automatically generates a skeleton of the .DAT and .cpp files identifying the states and controls. These files are later modified to define constraints, objective function and all other OCP settings required.

Additionally to the model, the physical limits of the controls and states are given to MUSCOD-II by means of (4.10) and (4.11) as requirements for a more accurate TM.

$$\begin{bmatrix} \boldsymbol{x}_{min} \\ -10\,^{\circ}\mathrm{C} \\ -10\,^{\circ}\mathrm{C} \\ -10\,^{\circ}\mathrm{C} \\ -5000\,\mathbf{W} \\ -10\,^{\circ}\mathrm{C} \\ -10\,^{\circ}\mathrm{C} \\ -10\,^{\circ}\mathrm{C} \\ -10\,\mathrm{kWh} \\ -10\,^{\circ}\mathrm{C} \\ -10\,^{\circ}\mathrm{C} \\ -10\,\mathrm{km/h} \\ -500\,\mathrm{Nm} \\ -10000\,\mathrm{rpms} \\ -120\,\mathrm{kW} \\ -10\,^{\circ}\mathrm{C} \end{bmatrix} \leq \begin{bmatrix} \mathbf{x} \\ T_{out_{VALVE2}} \\ T_{out_{VALVE4}} \\ T_{out_{VALVE1}} \\ \dot{Q}_{CHILLER} \\ T_{out_{CHILLER}} \\ T_{out_{CHARGER}} \\ T_{out_{PE}} \\ E_{BAT} \\ T_{out_{BAT}} \\ T_{out_{COOLER}} \\ v \\ M \\ n \\ P_{car} \\ T_{ambient} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{x}_{max} \\ 60\,^{\circ}\mathrm{C} \\ 60\,^{\circ}\mathrm{C} \\ 60\,^{\circ}\mathrm{C} \\ 5000\,\mathbf{W} \\ 60\,^{\circ}\mathrm{C} \\ 60\,^{\circ}\mathrm{C} \\ 60\,^{\circ}\mathrm{C} \\ 8.8\,\mathrm{kWh} \\ 60\,^{\circ}\mathrm{C} \\ 60\,^{\circ}\mathrm{C} \\ 200\,\mathrm{km/h} \\ 500\,\mathrm{Nm} \\ 10000\,\mathrm{rpms} \\ 120\,\mathrm{kW} \\ 60\,^{\circ}\mathrm{C} \end{bmatrix}. \tag{4.10}$$

This way it is quite straightforward to force that the maximal working temperature for the coolant in this circuit is 60°C. Additionally, to protect components from a sudden change in temperature minimum PWM constraints are fixed for the pumps control signals in order to assure coolant flow through the components at any time.

$$\underbrace{\begin{bmatrix} 0 \\ 10 \\ 16 \\ 0 \\ 0 \\ 30 \end{bmatrix}}_{\boldsymbol{u_{min}}} \leq \underbrace{\begin{bmatrix} Valve_{COOLER} \\ Pwm_{FAN} \\ Pwm_{BAT} \\ Valve_{CHILLER} \\ Valve_{CIRCUIT} \\ Pwm_{PE} \end{bmatrix}}_{\mathbf{u}} \leq \underbrace{\begin{bmatrix} 1 \\ 90 \\ 100 \\ 1 \\ 1 \\ 100 \end{bmatrix}}_{\boldsymbol{u_{max}}}. \tag{4.11}$$

These constraints are introduced in the .DAT file of Appendix B in the lines 92-128 and 192-210.
It is important to notice that by means of the temperature and the pump constraint, the PE failure is avoided.

4.2.3 OCP formulation

With all these requirements, the open-loop finite-horizon OCP associated to the cooling circuit can be formulated as follows:

$$\min_{x^*(\cdot),u^*(\cdot)} \Phi = \int_{t_0}^{t_0+\tau} c\,dt \tag{4.12a}$$

$$\text{s.t.} \quad 0 = x(t_0) - \overline{x}_0, \tag{4.12b}$$

$$\dot{x}(t) = f(x(t), u(t), p) \qquad \forall t \in H_p, \tag{4.12c}$$

$$x_{min} \leq x \leq x_{max} \qquad \forall t \in H_p, \tag{4.12d}$$

$$u_{min} \leq u \leq u_{max} \qquad \forall t \in H_p. \tag{4.12e}$$

Given an initial value of the differential states, $\overline{x}_0$, at time t_0, the goal of the strategy is to find the controls and resulting state trajectories $u^*(\cdot), x^*(\cdot)$, that minimize the objective function Φ and satisfy the constraints for a given prediction horizon of duration H_p. Notice that during the OCP performance possible disturbances d are not considered, as it will be seen later, this issue is addressed with the close of the open-loop feedback shown in figure 4.2. For this reason, hereafter disturbances d are left out from the notation.

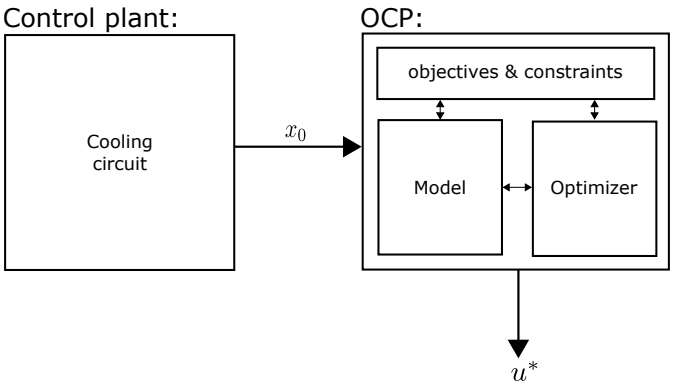


FIGURE 4.2: Optimal control problem outline.

4.3 Solving the OCP

For solving an OCP as the formulated in (4.12) there exist several methods as presented in [70]. In this section, first an overview of the methods is shown, where the focus is on their benefits/drawbacks and not on the mathematical details. Afterwards, the most popular alternatives, the direct methods, is further subdivided and finally the strategy behind MUSCOD-II based on the DMS method is described.

4.3.1 Methods overview

The various existing approaches for solving OCPs constrained to continuous time dynamic systems can be divided in the following categories:

- **Indirect methods:** find an optimal solution by satisfying optimality conditions and provide results with high accuracy. In contrast to the other approaches, they optimize in an infinite dimensional function space. The main drawback of indirect methods is that they are based on an interactive process that requires insight into the problem and paperwork for the specific instance to be solved and which cannot be automated.
- **Dynamic Programming (DP):** solves the problem breaking it down into a set of simpler subproblems that are solved just once and whose solution is stored. The idea is to reuse the information stored, each time a previously computed subproblem occurs. The advantage of DP is that it can find a global optimal solution thanks to an entire state space search. Nevertheless, the technique suffers from the "curse of dimensionality" which leads to unacceptable slow run times in big problems.
- **Direct methods:** are the most popular and promising in literature. These methods discretize the infinite horizon problem to transform it into a suitable form for numeric

solution by a nonlinear optimization solver. Although they do not lead to a global optimum, there exist several strategies based on direct methods that guarantee a small optimality loss and offer in addition a flexible problem formulation and fast run times.

Given the large model presented in Chapter 3 and being the final goal of the research to perform the control online, the direct method approach was the most suitable. For this reason, the optimization package MUSCOD-II which is based on a direct method: the DMS was chosen.

4.3.2 Direct methods

Direct methods aim at solving the OCP (4.4) numerically and therefore are based on the discretization of the problem in an N-points grid :

$$\min_{x,u} \quad \sum_{i=1}^{N-1} l_i(x_i, u_i, p) \tag{4.13a}$$

$$\text{s.t.} \quad 0 = x_0 - \overline{x}_0 \tag{4.13b}$$

$$0 = x_{i+1} - f_i(x_i, u_i, p) \qquad i = 0, ..., N-1, \tag{4.13c}$$

$$0 \leqslant c_i(x_i, u_i, p) \qquad i = 0, ..., N. \tag{4.13d}$$

where l_i, f_i, c_i stand for the discretization of the cost function, ODEs system's right-hand side and constraints, respectively.
Depending on the way they outline the discrete problem (4.13) and reformulate it in the nonlinear problem form, direct methods can be divided in two main groups. For further categorizations depending on other major algorithmic components, the reader is referred to [71]:

- **The sequential approach:** sees the state trajectory as a dependent value of the controls $u(\cdot)$ and an ODEs solver is used to solve the IVPs formed by the constraints (4.4b) and (4.4c). This way, the discretized states x_i disappear from (4.13) and thus the number of unknowns of the non linear program is reduced to the discretized controls u_i. An iteration in these methods requires two sequentially performed steps (hence the name): system simulation and optimization. One of the first and most known sequential methods is the Direct Single Shooting (DSS).

- **The simultaneous approach:** addresses the full nonlinear program of equation 4.13 with the states and the controls as unknowns applying a Newton type optimization

algorithm. In these methods, optimization and simulation are performed simultaneously. Most popular methods are the Direct Collocation and the Direct Multiple Shooting (DMS).

Although the sequential approach has much less variables, the simultaneous has more structure in the linear subproblems and moreover, as stated in [71], the Newton type optimization for the simultaneous methods behaves with faster local convergence rates, specially for unstable or highly nonlinear systems, since as the authors intuitively describe "the nonlinearity is equally distributed over the nodes".
Nevertheless, sequential methods are often used since they have a simpler implementation and the idea is easy to grasp [70].

4.4 MUSCOD-II methods

4.4.1 The Direct Multiple Shooting (DMS)

The optimization tool used in this thesis, MUSCOD-II, is based on the direct multiple shooting. The idea of the multiple shooting is to discretize the controls inside a shootings grid with N intervals:

$$t_0 = \tau_0 < \tau_1 < ... < \tau_N = t_f \tag{4.14}$$

with the help of the following piece-wise constant functions:

$$u(t) := q_i, \qquad t \in [\tau_i, \tau_{i+1}), \tag{4.15}$$

where $q_i \in \mathbb{R}$. Additionally at each grid point, the node values s_i are defined to be used as initial values for the resulting N independent IVPs. This can be seen in the left plots of figure 4.3, where the discretization with five shooting points of a control variable $u[j]$ and the four IVPs with initial values $s_0, s_1 ... s_3$ for a state $x[k]$ are shown in the bottom and top plots, respectively.

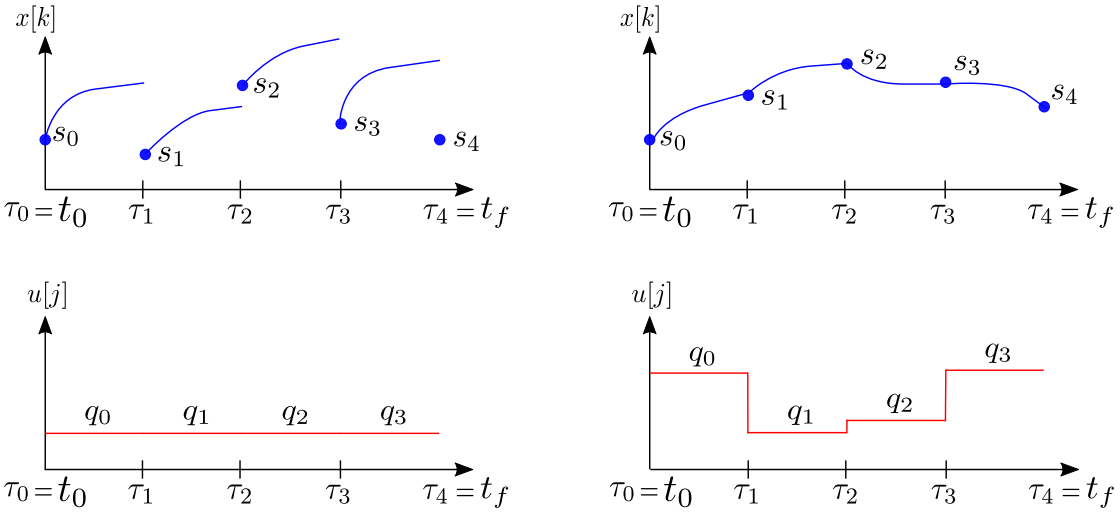


FIGURE 4.3: Direct Multiple Shooting before (left plots) and after (right) convergence is achieved.

Additionally to the mentioned advantage of faster local convergence due to the distribution of nonlinearities over the intervals, the fact of having separate IVPs that can be solved independently in parallel can lead to a further decrease in computation time using, e.g, multicore systems.

As it can be seen in the top plot on the left of figure 4.3, the problem of solving various IVPs separately is that the resulting entire state trajectory must not be necessarily continuous. In order to assure continuity, the DMS introduces the following constraints:

$$s_{i+1} = x_i(\tau_{i+1}; \tau_i, s_i, q_i, p), \qquad 0 \leq i \leq N-1, \tag{4.16}$$

where the right-hand side denotes the solution of the IVP on the shooting interval i, evaluated in τ_{i+1}, and depending on the initial time τ_i, initial state s_i, controls q_i and model parameters p. These matching conditions are allowed to be infeasible during the iterations, but are driven to feasibility once the numerical method has converged, as it can be seen in the right plots of figure 4.3.

Consequently, the nonlinear program problem resulting from the DMS can be formulated as:

$$\min_{w} \sum_{i=0}^{N} l_i(\tau_i, s_i, q_i, p) \tag{4.17a}$$

$$\text{s.t} \quad s_{i+1} = x_i(\tau_{i+1}; \tau_i, s_i, q_i, p), \qquad 0 \leq i \leq N-1, \tag{4.17b}$$

$$0 \leq c(\tau_i, s_i, q_i, p), \qquad 0 \leq i \leq N, \tag{4.17c}$$

$$0 = s_0 - x_0. \tag{4.17d}$$

where $w = (q_0, q_1 ... q_{N-1}, s_0, s_1, s_2 ... s_{N-1})$ is a vector with all the unknowns: the control parameters and the node values. For solving the IVPs in the N intervals, MUSCOD-II uses the ODE/ DAE solver DAESOL [72] combined with the Sequential Quadratic Programming (SQP) method that will be roughly described as follows.

4.4.2 Sequential Quadratic Programming

The NLP in (4.17) can be written as:

$$\min_{w} \quad F(w) \tag{4.18a}$$

$$\text{s.t} \quad G(w) = 0, \tag{4.18b}$$

$$H(w) \leq 0. \tag{4.18c}$$

where (4.17b) and (4.17d) are subsumed in (4.18b).
Under mild conditions, any locally optimal solution w^* of the problem, has to satisfy the Karush-Kuhn-Tucker (KKT) conditions:

$$\nabla_w \mathscr{L}(w^*, \lambda^*, \mu^*) = 0, \tag{4.19a}$$

$$G(w^*) = 0, \tag{4.19b}$$

$$H(w^*) \leq 0, \tag{4.19c}$$

$$H_i(w^*)\,\mu_i^* = 0, \quad i = 1, ... n_H. \tag{4.19d}$$

where $\mathscr{L}(w^*, \lambda^*, \mu^*) = F(w^*) + G(w^*)^T\,\lambda^* + H(w^*)^T \mu^*$ is the Lagrange function with multipliers λ^*, μ^* and n_H is the number of inequality constraints.
To solve the KKT system MUSCOD-II uses a Newton type optimization method, the Sequential Quadratic Programming (SQP). The SQP tries to find iteratively the points that satisfy (4.19) linearizing all nonlinear functions. It appears that, fortunately, the resulting linear complementarity system can be interpreted as the KKT conditions of the QP bellow:

$$\min_{\Delta w_k} \quad \nabla_w F(w_k)^T + \frac{1}{2} \Delta w_k^T \nabla_w^2 \mathscr{L}(w_k, \lambda_k, \mu_k) \Delta w_k \tag{4.20a}$$

$$\text{s.t} \quad G(w_k) + \nabla_w G(w_k)^T \Delta w_k = 0, \tag{4.20b}$$

$$H(w_k) + \nabla_w H(w_k)^T \Delta w_k \leq 0. \tag{4.20c}$$

Starting with an initial guess w_0 provided by the user, the SQP algorithm iterates $w_{k+1} = w_k + \alpha_k \Delta w_k$ with step direction Δw_k and step length α_k until a desired convergence criterion is satisfied. At each iteration not only the functions but also the Jacobians ($\nabla_w F, \nabla_w G, \nabla_w H$) and the Hessian matrix ($\nabla_w^2 \mathscr{L}(w_k, \lambda_k, \mu_k)$) have to be evaluated. Together with the solution of the QP problem, this is often the most crucial and expensive step, from the computation time point of view. An advantage of the DMS parameterization in the QP problems solution is that they have a sparse block structure (4.20) that can be exploited by means of a condensing algorithm to reduce the effort of the DMS solution to the same order as for the single shooting [54]. Moreover, concerning the derivatives calculation, it must be said that the block structure of the Hessian due to the DMS can be exploited with the high rank formula also presented in [54] in order to improve the convergence rate.

Chapter 5

Nonlinear Model Predictive Control: Simulation Results

"However beautiful the strategy, you should occasionally look at the results."

— Winston Churchill

CONTENTS:

In this chapter, after a brief introduction to NMPC and a description of the MUSCOD-II numerical methods for real-time performance, the NMPC of the cooling circuit is prepared in a simulation environment. The same model, as the one described in Chapter 3 and contained in MUSCOD-II, is used as controlled plant in a Software In The Loop (SIL). With this configuration, the different relevant parameters of the optimizer are adjusted to obtain an efficient and accurate TM. Furthermore, several driving cycles are studied and discussed to validate the results, prior to implement the real-time control in the vehicle.

5.1 Theory behind NMPC

Since the NMPC model will not, in general, capture the system behavior with 100% accuracy and furthermore the dynamic process is subject to disturbances, the control loop must be closed, as shown in figure 5.1:

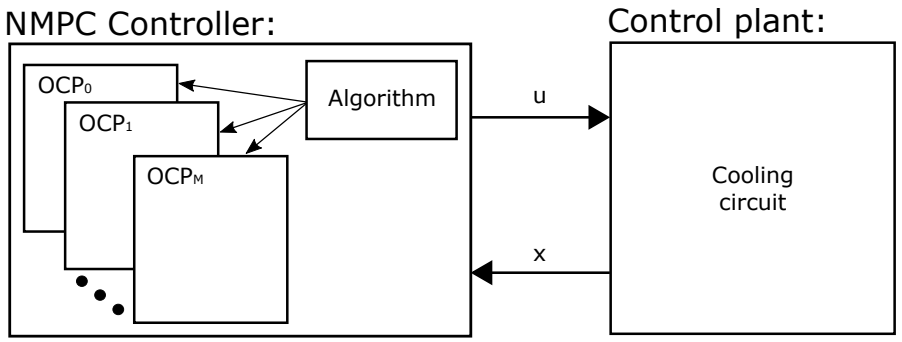


FIGURE 5.1: Nonlinear Model Predictive Control loop.

As it can be seen in figure 5.1, NMPC consists of several OCPs. How these OCPs are formulated and solved during the dynamic process can be understood with the conceptual example represented in figure 5.2, where the goal of the control is the tracking of a certain reference trajectory (in green) inside a given optimization horizon H_p split in four shooting intervals. Moreover, for better comprehension, it is assumed that the system has only one single control and state variables (in red and blue, respectively).

The top plot of figure 5.2 shows the situation at a certain time instant t_k, where the previous control/state signals exchanged between controller and plant are drawn with solid lines.
At instant t_k, an OCP (OCP_0) is formulated using the current state of the plant x_k for the initial condition (4.4b). To solve OCP_0, the numerical methods described in Chapter 4 are launched to find the optimal control sequence q_j^0 with $j = 0, 1...3$ for the tracking of the reference trajectory inside H_p, assuming that the prediction of the model is accurate enough (blue dashed line).
Since the calculations to solve the numerical problem are not instantaneous, the last value of the control q_i is kept while the current OCP is being solved. This can be seen in the middle plot of figure 5.2, where the control trajectory (red solid line) has been extended with the last value q_i.
At this new time instant t_{k+1}, the state of the plant is measured, thus being now assessable how accurate the prediction of the state trajectory was. In this example, this is overstated with the yellow area in the middle plot of figure 5.2, where the model mismatches and

the effect of disturbances lead to notable discrepancies between the predicted and the real scenario.

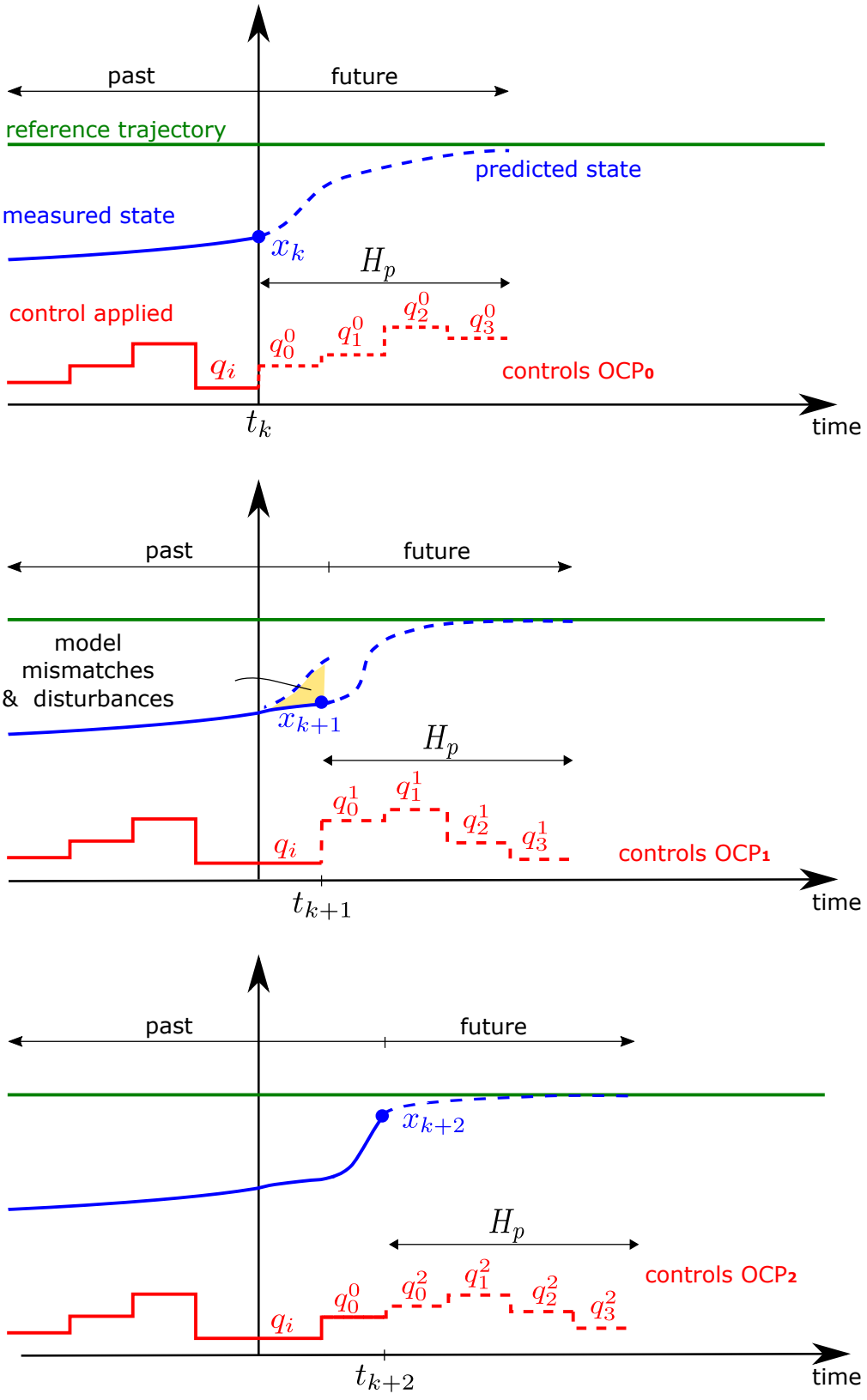


FIGURE 5.2: Nonlinear Model Predictive iterative process in a receding horizon fashion.

To mitigate this negative impact, a new OCP (OCP_1) is formulated and solved. During its computation, again the first step of the optimal control q_0^0 computed in the previous instant t_k is taken, as it can be seen in the red solid line of the bottom plot of figure 5.2. The rest of the control trajectory calculated for the $OCP_0(q_1^0, q_2^0, q_3^0)$ is discarded. Notice that the control q_0^0 is taken with one sample delay.
Finally, as indicated in the bottom plot, a new OCP (OCP_2) is formulated and solved. The process would continue as described shifting the prediction horizon over time, being this the reason for NMPC being also called Receding Horizon Control.

5.2 Real-time algorithms

One of the main bottlenecks in NMPC automotive applications is to solve the sequence of arising optimization problems in real-time. While the linear approach of MPC falls in the category of convex programming and hence local solutions are also global solutions, in the nonlinear version this is, in general, not the case [66]. As a consequence, NMPC faces the dilemma of either stopping the iteration procedure after a pre-specified convergence criterion is satisfied or prematurely according to a pre-specified computation time limit. The first option leads to considerable feedback delays, threatening real-time performance, and the second one to an approximate solution which may imply a large optimality loss.
Since both criteria, computation time and optimality loss, are crucial for successful online NMPC, the state-of-the-art algorithmic ideas collected in [71] aim at reducing both at the same time.

5.2.1 Real-time iteration scheme

Among the various existing methods, in this research the Real-Time Iteration (RTI)-scheme [73] was used. This method introduces the following modifications in the OCP (4.12) solution described in Chapter 5:

- **Single iteration:** With the RTI-scheme only one SQP iteration is performed at each sampling time. Although this fact means that the controlled plant receives only an approximate of the optimal feedback control, reacting enough fast to possible disturbances is preferable to the alternative of spending large times calculating with outdated values. On the other hand, subsequent OCPs differ little from each other and therefore solution information of the previous problem can be exploited to initialize the next one. This transition is done in such a way that after an initial disturbance, it subsequently

delivers approximations for the optimal feedback control that become better and better if no further disturbance occurs [74].

- **Embedding strategy:** The initial states of the controlled plant only appear in the OCP in the form of (4.12b) and through the DMS discretization, it becomes the linear constraint (4.17d) of the resulting NLP. Hence, subsequent OCPs differ only in the initial states constraint. Then it can be said that the initial states are embedded as linear constraints in the OCP. The RTI exploits this fact to initialize the OCP at a sample time t_k with the solution obtained for the previous OCP at t_{k-1}. Although this implies a violation of the constraint (4.17d), due to its linearity, this is immediately corrected after the first (full) SQP iteration.

- **Feedback response and preparation phases:** Thanks to the embedding strategy, the SQP steps can be performed separately in two phases according to their need or not of current state information. The two phases are the feedback response FRP, in green, and the preparation phase PP, in yellow, in figure 5.3, respectively. The main advantage of this method is that given a new measurement of the current state of the plant x_{k+1}, a fast response with the control signals u_{k+1} can be given after a short FRP_{k+1}. During the FRP_{k+1}, the current state and the previously prepared information calculated in PP_{k+1} are used to solve an OCP. Notice in figure 5.3 that the calculations performed in the FRP are orders of magnitude shorter than the PP and that the duration of both phases are variable for each OCP.

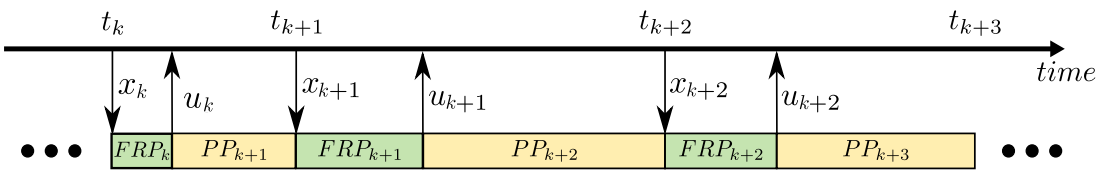


FIGURE 5.3: Nonlinear Model Predictive Control loop taken and extended from [75].

It must be highlighted that while in the vehicle the states-control exchange between controlled plant and controller takes place asynchronously as shown in figure 5.3, the simulation tests are realized with a synchronous exchange.

5.2.2 Robustness and stability RTI-scheme

The stability of the closed loop system, resulting from the RTI-scheme implemented in MUSCOD-II, is addressed by combining concepts from both NMPC stability theory and convergence analysis of Newton-type optimization methods in [76]. Given that the sense of this thesis is the application of a previously-developed control strategy in the optimization package MUSCOD-II, the deep features related to the NMPC performance such as robustness and stability are out of the scope. However, we are aware of the inherent robust properties of the considered control technique (given the dynamic optimization behind the approach) and the stability of the closed loop given the feasibility of the multi-objective optimization problem under certain conditions [77].

5.3 Software in the loop for NMPC

Before implementing NMPC in the real vehicle, the method was validated in the simulation environment of figure 5.4.

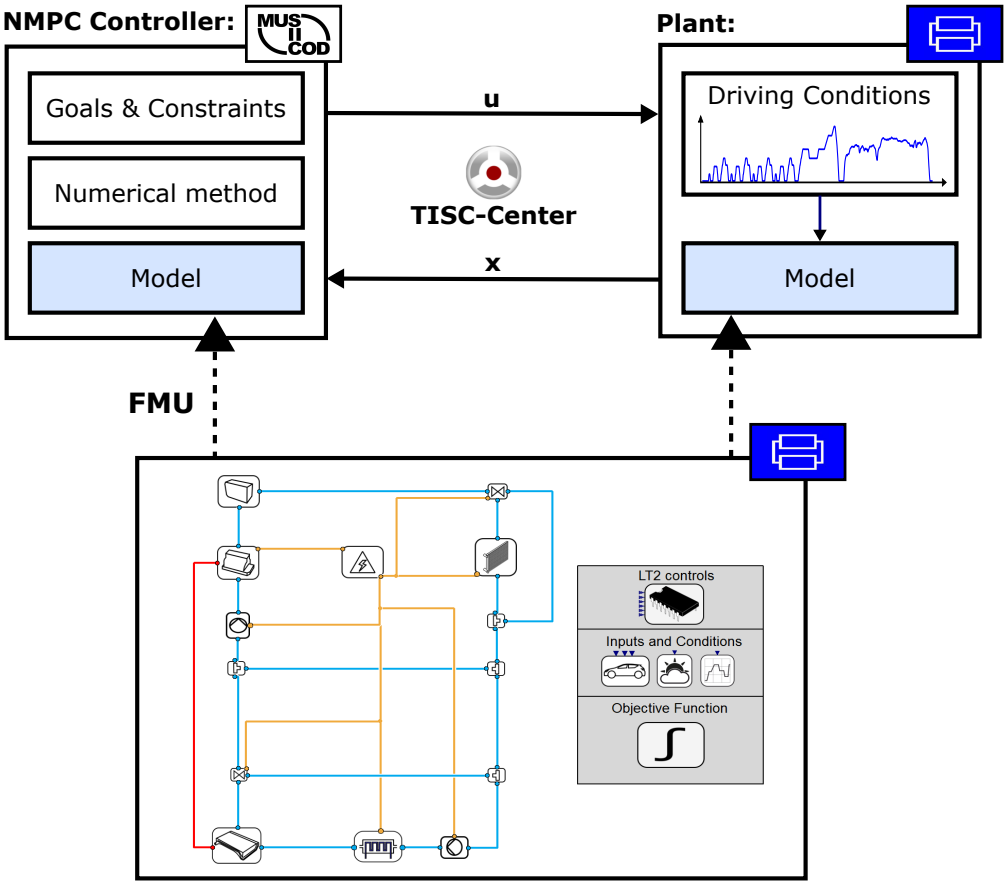


FIGURE 5.4: Software in the loop for NMPC. The controller in MUSCOD-II and the plant in Dymola (blue symbol) are coupled by means of the co-simulation tool TISC-Center, product of TLK-Thermo GmbH.

Here, the same Dymola model exported as FMU to describe the ODE system to the optimizer, was used as controlled plant. The only modification in this model was the inclusion of the driving conditions for different test cycles. This information was stored in look-up tables with the time as input and the traction variables M, n, P_{Car} and v as outputs. Moreover, the ambient air and initial coolant temperatures were given constant values along the cycle duration.
With these modifications, the model of the plant could provide all the differential states required to the NMPC controller being the only inputs of the model, the controls of the actuators.
For the communication between MUSCOD-II and Dymola, the co-simulation software TISC-Center from TLK-Thermo GmbH [78] was chosen due to its efficiency in reducing the numerical inaccuracies associated with time-discrete variables exchange. It was decided that the synchronization rate of 2.5 seconds was suitable for the control of the studied thermal system inertia.

Although the development of a more detailed model would have been a closer previous step to the real implementation, the coupling of MUSCOD-II with the same model as the one in its bowels is still quite interesting. It allows to adjust the optimization parameters and evaluate in an early stage the results obtained with the method, being the model/plant mismatches set aside.

It is very important to highlight that in this research two main simplifications were taken:

- the traction variables remain constant along the prediction horizon and hence their derivatives are equal to 0.
- the solenoid valves control signals output by MUSCOD-II are rounded before entering the controlled plant, thus taking 0 when the value is under 0.5 and 1 in the opposite case.

The first limitation assumes that no future information about the driving cycle is available in advance in the vehicle. Since the challenge of developing an "on board cycle predictor" [21] is already a complex and exhaustive question itself, this was left out of scope in this research where the main effort was put in the NMPC method.

The second limitation implies that the controls given to the plant are, in general, not optimal. The clean way to deal with the optimization of systems with discrete controls is to formulate a Mixed-Integer Optimal Control Problem (MIOCP) and use the appropriate numerical methods

for solving it. Nevertheless, the methods for solving MIOCPs often imply the introduction of extra unknowns, thus leading to larger problems and hindering real-time performance. Given the already high number of controls governing the system treated in this thesis and the requirement of controlling the plant on real-time, the MIOCP approach was discarded. For more information about real-time capable MIOCP, the reader is referred to [70].

5.4 Classical approach: Finite-state machine for thermal management

To evaluate if the simulation results obtained with NMPC are reasonable or not, a rule-based TM strategy was modeled in Matlab/Simulink, from now on referred as the "classical" approach. The main blocks of this model are shown in figure 5.5:

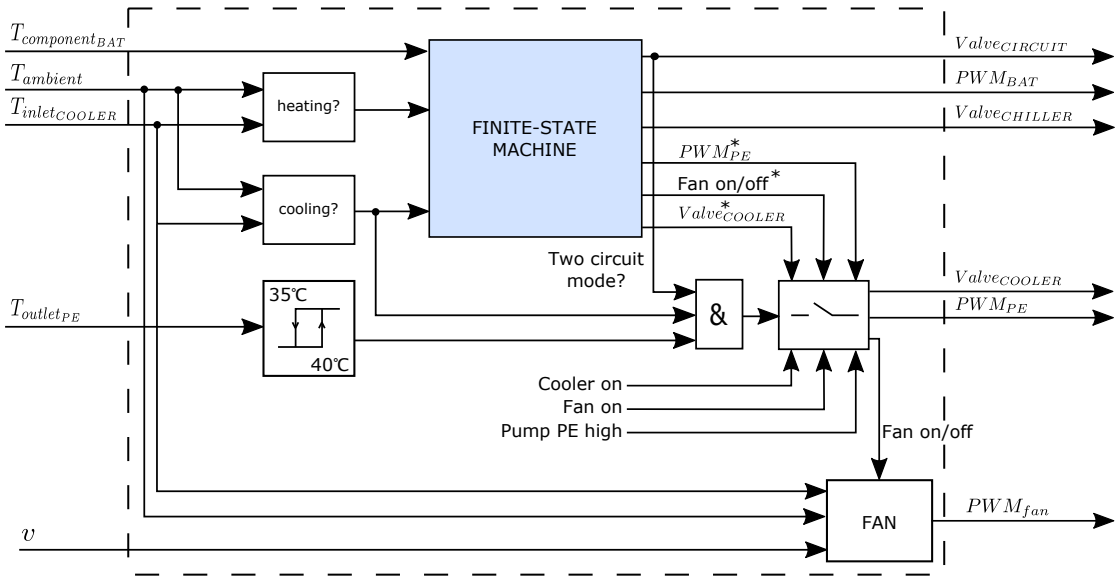


FIGURE 5.5: Classical approach for TM.

As it can be seen in figure 5.5, the inputs of this control system are the BAT, PE, cooler inlet and ambient temperatures and the vehicle speed. With the cooler and ambient temperatures it is possible to determine if the heating/cooling through the cooler is possible or not. This is calculated in the "heating?" and "cooling?" submodels of figure 5.5.

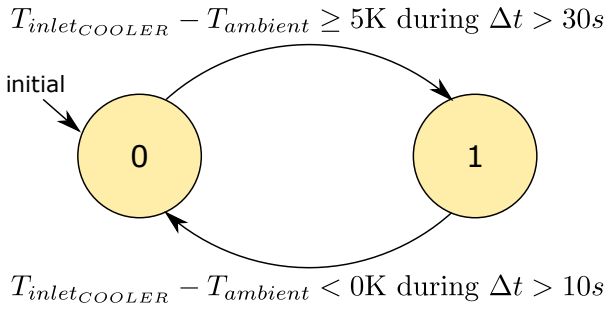


FIGURE 5.6: Finite-state machine for determining if it is possible to dissipate heat through the cooler or not.

Figure 5.6 is an insight in the "cooling?" submodel. Here, there are two possible states for the cooling variable: 0 if it is not possible to cool the circuit with the cooler and 1 otherwise. The transition from state 0 to state 1 is possible when the temperature difference between coolant and ambient is greater or equal than 5K during at least 30s. Once the cooling condition is satisfied, to return to the not cooling state, state 0, it is necessary that the temperature difference remains negative during more than 10 s.
The heating condition was developed analogously to the described cooling submodel.

The core of the control is a finite-state machine (blue box in figure 5.5) that uses the cooling/heating information and the battery temperature to calculate the controls PWM_{BAT}, $Valve_{CIRCUIT}$, $Valve_{CHILLER}$ and the signals PWM^*_{PE}, $Valve^*_{COOLER}$ and fan on/off*. The calculated controls are directly taken as outputs, the signals with the * not.
These signals are inputs in a switch box where the rest of inputs are the constant signals: cooler on, fan on and full PE pump power. This later cooling demand is only taken if the relay on the left in figure 5.5 is on, the cooling through the cooler is possible and the output of the finite-state machine concerning the circuit mode is two circuit mode. The relay switch on and off points are 40 and 35 °C, respectively. The main idea behind the switch is to protect the PE when the circuits are decoupled in case of high temperatures. In the one circuit mode, the PE shares the heat sinks used by the BAT and therefore it is already protected with the logic described in the finite-state machine.

On the other hand, the PWM signal for controlling the fan is calculated in the "FAN" submodel of figure 5.5 depending on the signal fan on/off, the vehicle speed and the coolant/air temperature difference in the cooler.
When the fan is on, the value of the PWM signal is calculated with the function represented in the figure 5.7. At low speeds, a high PWM signal for the fan is required to increase the

air flow rate while at high speeds this is not necessary. Concerning the ΔT in the coolant it can be said that it is worthless to use the fan (PWM_{fan}=10) when there is no temperature difference. The PWM demand increases over ΔT up to the threshold of 5K, from where it is assumed that the difference is large enough to use the highest PWM signal according to the speed.

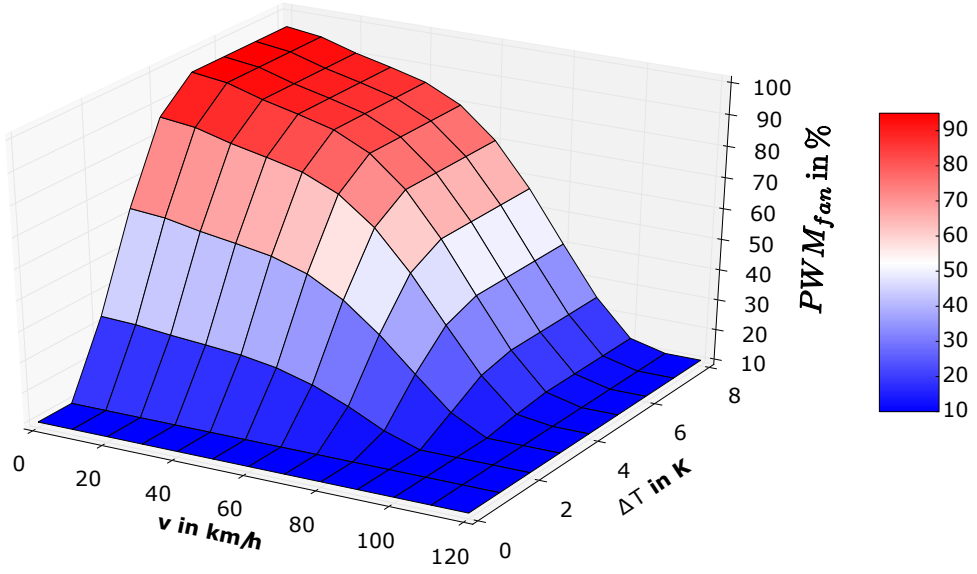


FIGURE 5.7: Fan function depending on the vehicle speed and coolant/ambient temperature difference in the cooler.

Finally, the finite-state machine can be seen in more detail in figure 5.8.

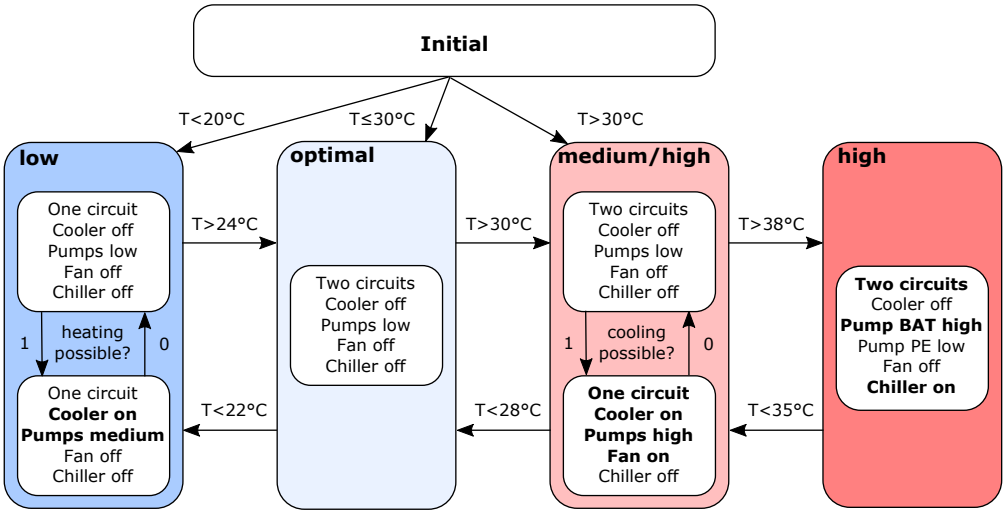


FIGURE 5.8: Finite-state machine used in the classical approach.

Depending on the BAT temperature, the cooling demand is assigned to one of the low, optimal, medium/high or high states. In the low state (darkest blue) the BAT is heated by means of the coupling with the PE circuit using the heat generated in the PE or the air, in case

it is warmer than the coolant. The optimal state (light blue) saves energy turning off the actuators. For medium/high temperatures (light red) the cooler and the fan are used to cool down the BAT while at temperatures above 38°C (dark red), the circuits are decoupled and the BAT is cooled down by means of the chiller.

The classical model was simulated together with the Dymola model of the controlled plant by means of the co-simulation tool TISC-Center with a synchronization rate of 2.5 s. The resulting configuration shown in figure 5.9 needs only 13 seconds for simulating 1000 s.

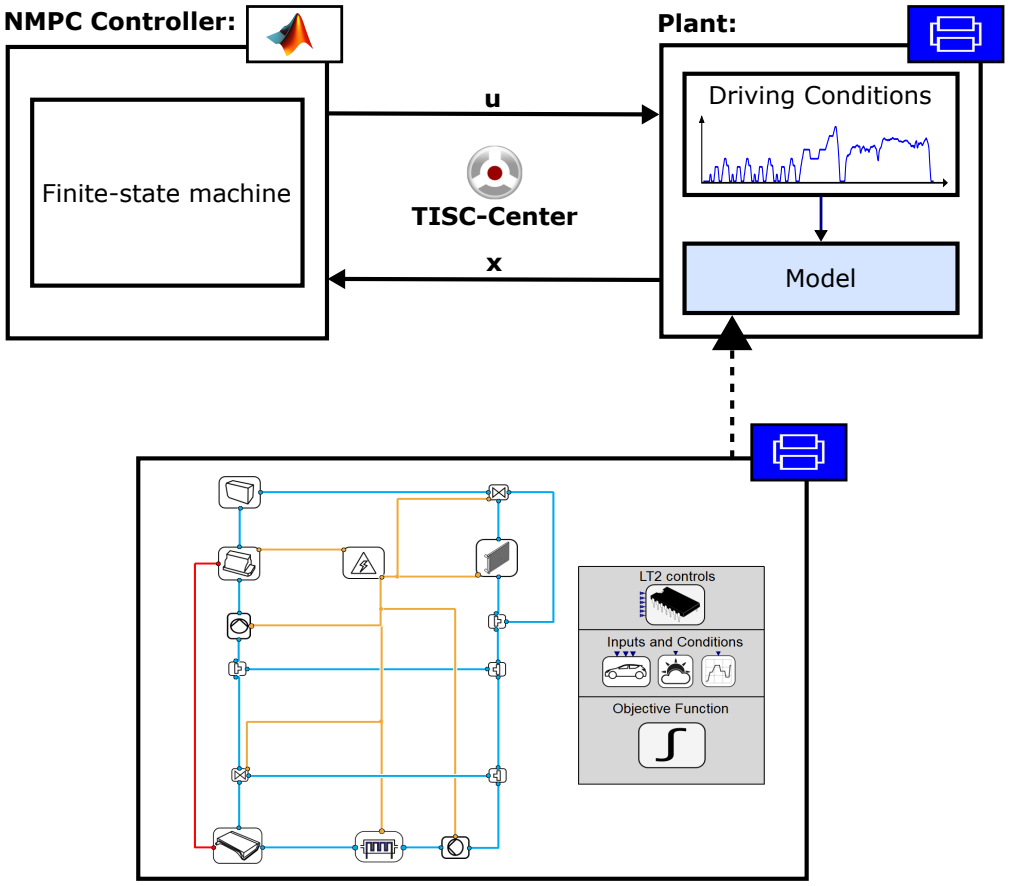


FIGURE 5.9: Software in the loop for the classical approach. The controller in Simulink and the plant in Dymola (blue symbol) are coupled by means of the co-simulation tool TISC-Center, product of TLK-Thermo GmbH.

Notice that while in the NMPC SIL, in figure 5.4, the x vector contains 15 variables needed for the prediction, the classical approach, in figure 5.9, needs only 5 states to calculate the controls.

5.5 Test scenarios

In order to compare the results obtained with the SIL configuration using the NMPC and the classical control, several test scenarios were designed. These scenarios consist of a driving cycle and the thermal conditions of the ambient and the coolant at the start of the trip.

5.5.1 Driving cycles

A driving cycle is a series of data points representing the speed over time profile of a vehicle. Driving cycles are usually designed and used to measure, under repeatable conditions, vehicle performance characteristics such as fuel consumption or exhaust gases emissions.

The most popular driving cycle in the European Union is the New European Driving Cycle (NEDC) since it is used for type approval of light-duty vehicle models. It can be seen in the blue box of the bottom plot in figure 5.10.The NEDC consists of a low speed cycle repeated four times representing an urban zone, followed by a higher speed section emulating a highway driving. The constant speed and acceleration periods in the NEDC make it repetitive but quite unrealistic compared to a real driving pattern.

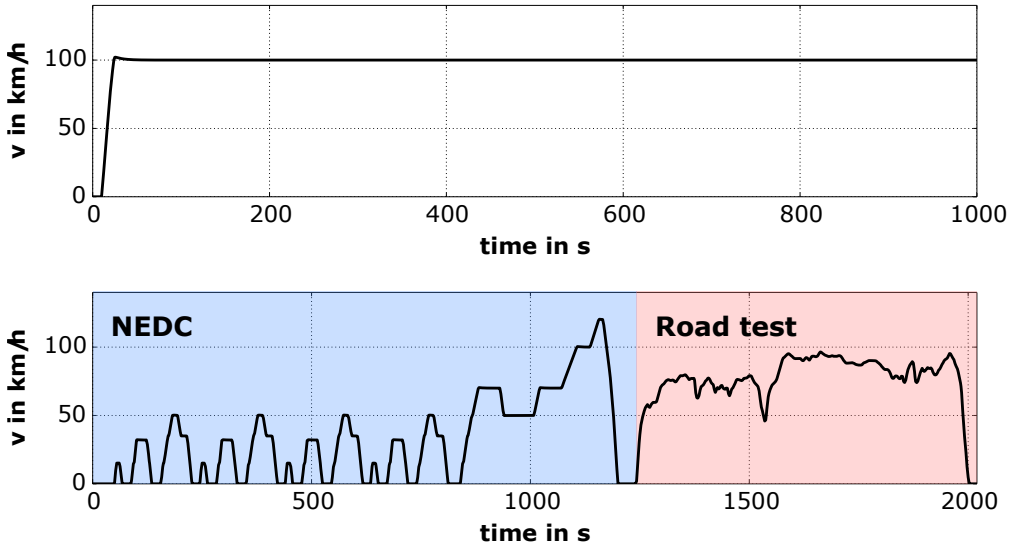


FIGURE 5.10: Studied driving cycles: A constant speed drive in a highway (top) and the EMPA B cycle (bottom).

In this thesis, an extended version of the NEDC was used: the EMPA B cycle taken from [65] and originally designed by the Swiss Federal Laboratories for Materials Science and Technology (EMPA) institute which is part of the Swiss Federal Institutes of Technology (ETH). The EMPA B cycle includes an extra highway section after the NEDC which comes from real

vehicle measurements as it can be seen in the red area of figure 5.10. Its total distance and time duration are 27.53 km and 2020 s, respectively and the average driving speed is 56.85 km/h. In the top plot of figure 5.10, the other driving cycle used for the simulations can be seen. It consists of a 0-100 km/h acceleration in 30 s followed by a constant 100 km/h drive which lasts 1000 s.

The aim of using the driving cycles described is to analyze the behavior of the NMPC compared to the classical approach in stationary and transitory scenarios.

5.5.2 Ambient and initial conditions

The driving cycles described will be analyzed under the different climate conditions presented in figure 5.11:

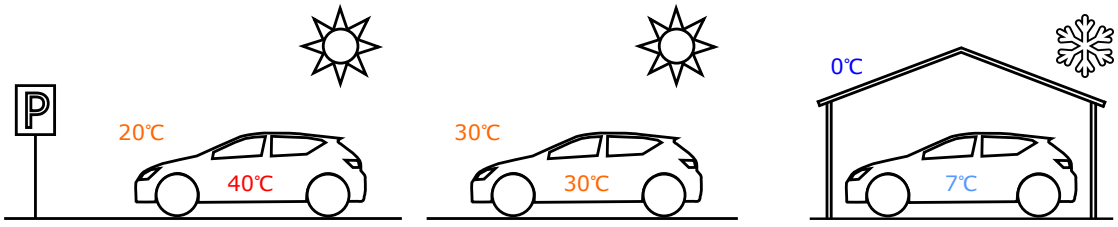


FIGURE 5.11: Hot(left), mild(middle) and cold(right) climate conditions used in the simulations.

The situation on the left in figure 5.11 represents an usual initial driving situation during hot summer days in a southern European country. Here the vehicle has been parked during the warmest hours in an open-air street under direct solar radiation. The effect of the solar radiation and the heating of the coolant during the charging process lead to higher temperatures in the coolant and the components, 40°C, compared to the ambient temperature, 20°C, when the driver parks out several hours afterwards.
The middle situation in figure 5.11 belongs to a mild scenario where the ambient and the initial coolant temperatures are 30° C.
The final thermal scenario stands for a common initial driving situation during winter in the northern European countries. The vehicle is parked out after having been the whole night in the garage. Therefore the initial coolant temperatures are higher, 7° C, than the ambient temperature of 0°C.
Finally it must be said that in the three cases, it is assumed that the ambient temperature remains constant and that the initial state of the BAT is fully charged.

5.6 Simulation results

Before analyzing the different results obtained for the NMPC and the classical control approach in the presented scenarios, the main highlights of the optimizer calibration will be given.

5.6.1 Optimization parameters settings

In the numerical methods used by MUSCOD-II, the number of shooting points and the length of the horizon must be provided by the user and have a significant impact in the solution and the computation time.

Therefore, several simulations were performed ranging the number of shooting points from 1 to 8 and the optimization horizon length from 100 to 800 seconds. The driving cycle taken for these simulations was the EMPA B under hot conditions. The effect of these parameters on the total cost at the end of the cycle $\Phi(t_{end})$ and the maximal computational time t_{SQP} observed for an SQP iteration are shown in figure 5.12.

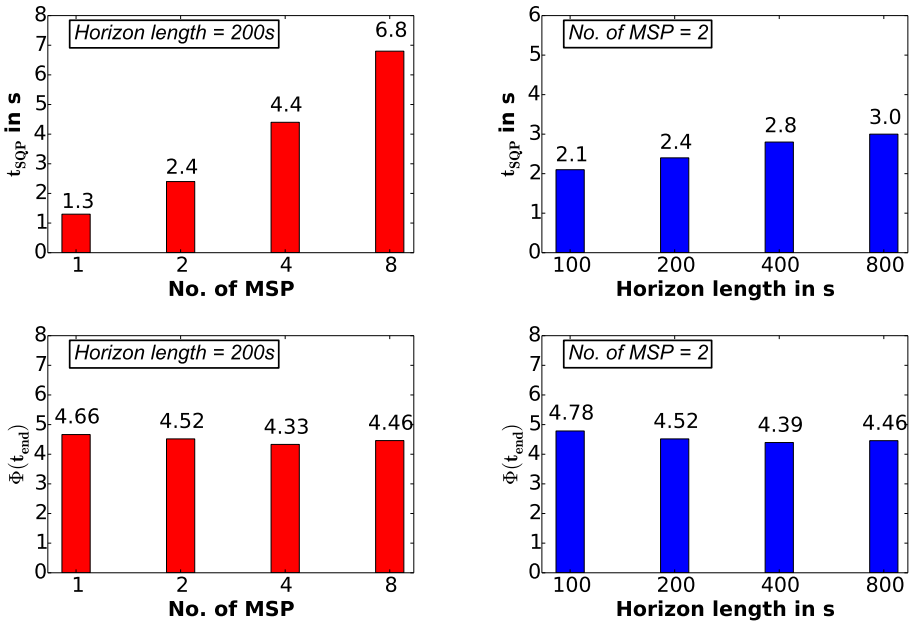


FIGURE 5.12: Effect of the number of multiple shooting points and the length of the optimization horizon on the total costs for the entire cycle Φ and on the maximal observed computational time needed for an SQP iteration.

As it can be seen in the top red bars of figure 5.12, the number of shooting points increases substantially the computational time. The reason for this is that the total number of unknowns in the NLP increases with the number of shooting points N according to $(N+1)\,n_x + N\,n_u$, where n_x and n_u are the number of states and controls, respectively.
Moreover, the effect of the number of shooting points on the total cost Φ shows a tendency, as it can be seen in the bottom red bars of figure 5.12. In general, the thinner the grid, the lower the observed total cost is. The only exception is the total cost with four shooting points, which surprisingly shows a lower total cost than the eight shooting points case. This can be due to the uncertainty introduced by the rounding of some control variables.

Concerning the prediction horizon, the blue bars in the top plot of figure 5.12 show that the longer it is, the more time is needed for computation. Nevertheless, it must be said that this effect is less influential than the number of shooting points.

Less significant is the effect on the total cost Φ, blue bars on the bottom of figure 5.12, where the minimum value is observed with a prediction horizon of 400 s. In general, the longer the prediction horizon is, the better the results are. However, given that the model predictions in this thesis do not include future information of the driving cycle, which is crucial for building a realistic scenario, long horizons are not necessarily synonymous with better results.

In order to reach a fast performance and costs reduction trade-off, an horizon of 200 s and a grid with 2 shooting points was fixed for the rest of this thesis. On the one hand with these values online performance is assured, being the maximal computation time of 2.4 s lower than the 2.5 s synchronization rate between controller and plant and on the other hand, the potential loss is assumable: the total costs obtained are only 4% higher than in the case with 200 s horizon and 4 shooting points (4.52 instead of 4.33).

5.6.2 Cost function parameters settings

Another key parameters for the correct optimizer performance are the costs function parameters $coef_T$ and $coef_P$ as introduced in (4.9). According to these parameters, the optimizer will give more value to minimize one goal than the other. In figure 5.13, the NMPC simulations with four different $coef_T/coef_P$ sets with $coef_T = 100$ and $coef_P$ ranging from 0 to 10, are compared to the classical control simulation, red lines in the figure. The results of each control strategy can be evaluated in terms of temperature, middle plot in figure 5.13, and electrical consumption, bottom plot in figure 5.13. In the temperature plot, the different health regions for the BAT are shown in different colors. Again the used driving cycle was

the EMPA B under hot conditions.

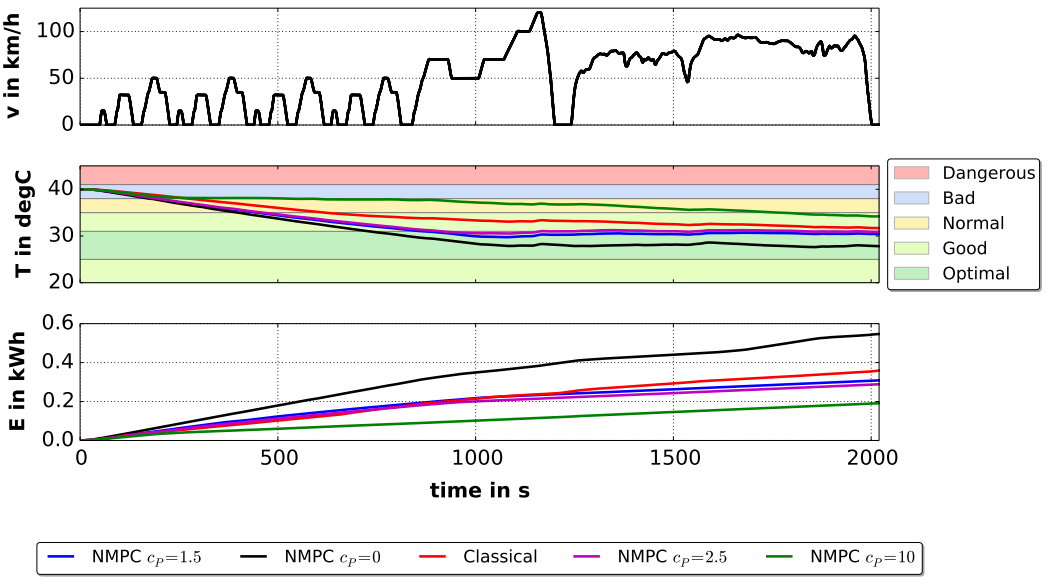


FIGURE 5.13: Effect of the objective function parameters on the TM.

As it can be seen with the black lines in the middle and bottom plots of figure 5.13, when the $coef_P$ is taken equal to zero, the BAT is driven to the optimal temperature disregarding the electrical consumption. At the other extreme, given a $coef_P$ of 10, green lines, the optimal temperature range is not reached in the whole driving cycle in order to achieve a considerable electrical consumption reduction.

Compared to the classical approach, red lines, the variants $coef_P$=1.5 and $coef_P$=2.5, in blue and magenta respectively, show a reduction in the electrical consumption together with a better temperature trajectory. Hence, both are suitable for this driving situation. The decision of choosing one of them depends on the preference of saving electrical energy or improving the temperature.

The same tuning procedure was carried out for the rest of the test scenarios leading to Table 5.1, which contains the parameters used in the simulations that will be presented in the next sections. Compared to a conventional PID tuning process, the cost parameters calibration is more straightforward. For example, given a scenario where the NMPC consumption results are poorer than expected, the $coef_P$ can be increased and the results will show an improvement in this goal achievement. On the contrary, during the PID tuning, the effect of the P, I and D gains on the several goals is not so intuitively and directly attributable to them.

TABLE 5.1: Cost function parameters for the studied scenarios

Conditions / Cycle	Hot		Mild		Cold	
	$coef_T$	$coef_P$	$coef_T$	$coef_P$	$coef_T$	$coef_P$
EMPA B	100	1.5	100	1.5	100	0.5
Constant	100	2.5	100	4	100	1.5

The calibration of the cost function parameters is a necessary step in NMPC and in this sense a simulation validation at an early stage can be very helpful to build look-up tables storing the necessary parameters for different driving situations.

5.6.3 EMPA B under hot conditions

In figure 5.14 the achievement of the TM goals with the NMPC and classical approaches for the EMPA B driving cycle under hot conditions is shown. As explained before, in contrast to the classical control, in red, the NMPC, in blue, reaches the "optimal" temperature range and at the same time leads to a reduction in the total electrical consumption of the actuators. How this is achieved can be understood taking a look at the controls of both strategies in figure 5.15.

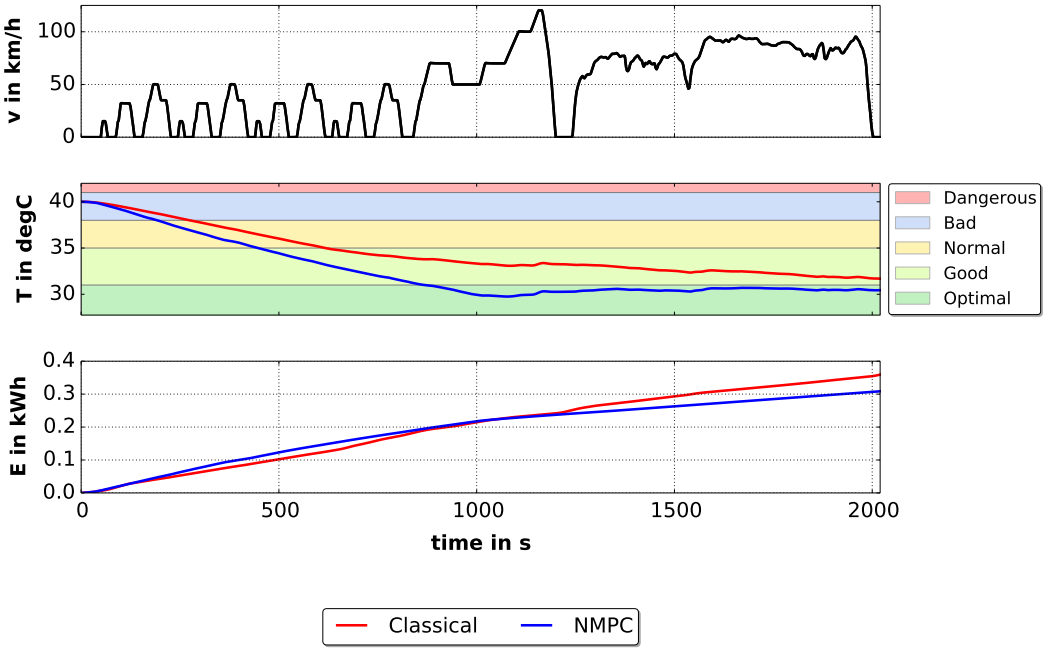


FIGURE 5.14: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the EMPA B driving cycle under hot conditions.

The classical strategy presents two clear working points as it can be seen in the red line in the top middle plot of figure 5.15, where during the first 625 s the BAT and PE circuits are decoupled (Circuit valve = 0). In this configuration, the BAT is cooled down with the chiller, as it can be seen in the top right plot (Chiller valve = 1). At the same time in this interval, the PE is cooled down with the cooler (Cooler valve = 0) and fan, top and bottom left plots, as soon as the cooling condition is satisfied and till the component reaches the hysteresis temperature of 35°C at time 210 s. Once the BAT reaches the limit of 35°C, the cooling through the chiller is deactivated and the strategy turns to the one circuit mode to use the cooler, the fan and the pumps to cool down the two components.

On the contrary, the NMPC strategy couples the two circuit from the beginning, switching to the decoupled mode during short intervals, and combines the use of the chiller and the cooler at the same time. Around the time 1000 s, the chiller is deactivated, since the BAT has reached the optimal range, as it can be seen in the solid blue line in figure 5.16. Compared to the classical strategy, the combination of the simultaneous usage of the two heat sinks together with the pumps and fan rationing lead to the temperature profile and consumption improvement.

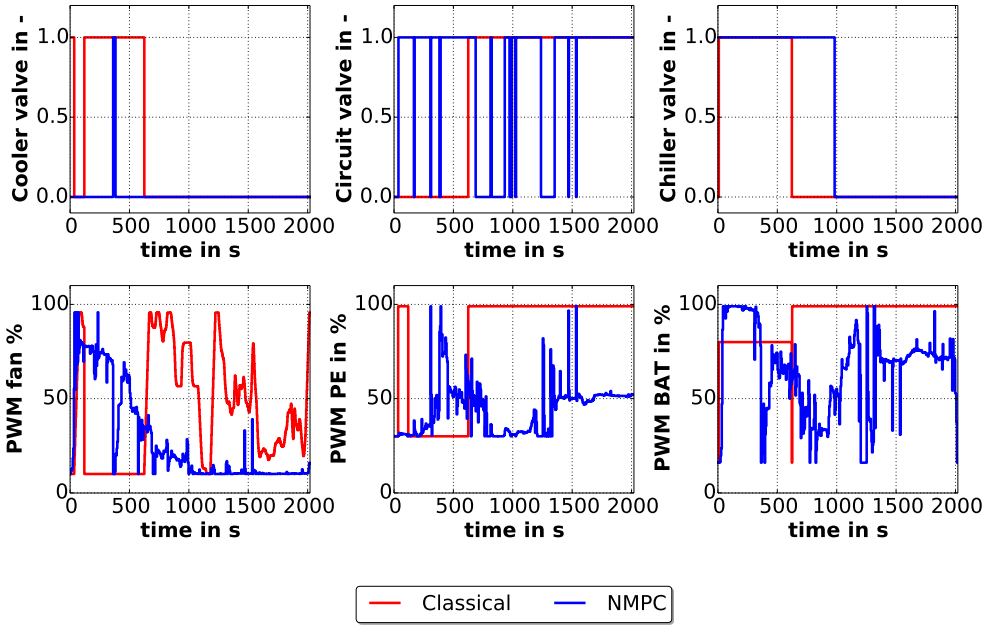


FIGURE 5.15: NMPC vs Classical TM controls for the EMPA B driving cycle under hot conditions.

Figure 5.16 shows the temperatures of the electric components BAT and PE in blue and red, respectively and the ambient temperature in black for the NMPC, solid lines, and the classical approach, dotted lines. The effect of the two circuit mode during the first 625 s in the classic

control leads to the different temperature levels of the red and blue dotted lines.
The main highlight of this figure is the significant gap between the components temperatures and the ambient air which provides the cooler with a high cooling potential. The NMPC strategy captures this fact and uses it from the first moment thus achieving lower temperature levels in both components.

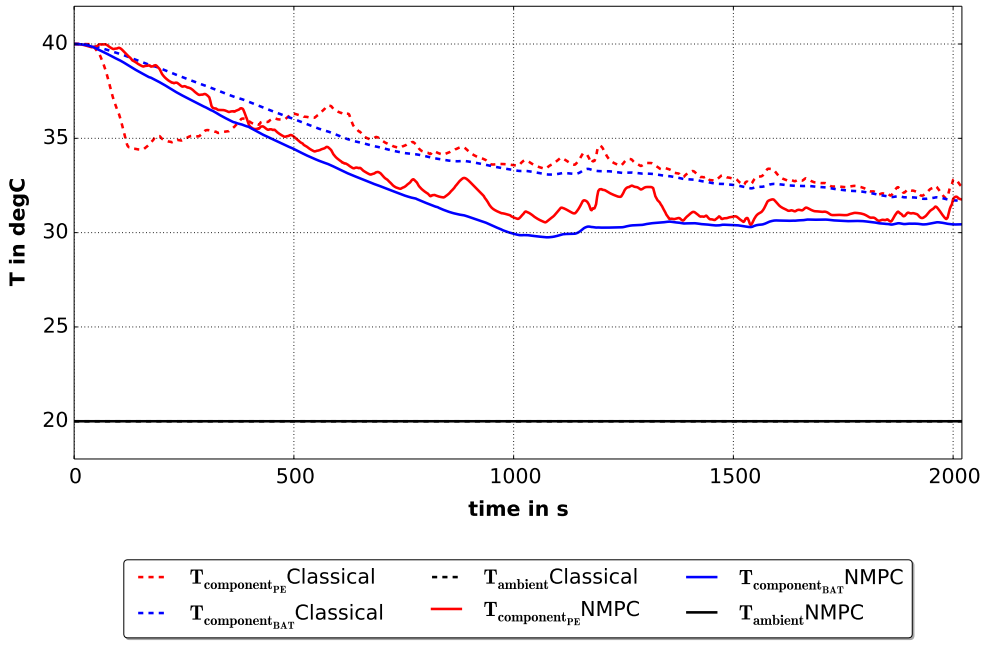


FIGURE 5.16: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the EMPA B driving cycle under hot conditions.

The main differences of both strategies can be also interpreted from the costs point of view, shown in figure 5.17. Since NMPC exploits the cooler potential from the first moment, the slope of the total costs c, blue line in the bottom plot of figure 5.17, decreases faster than in the classical control, red line. Nevertheless, in the first 625 s, as it can be seen in the middle plot of this figure, NMPC invests more than the classical control in reducing the temperature costs paying a higher cost in the consumption.

All in all, it can be said that NMPC shows a better TM, achieving 14.12% consumption reduction and an average BAT temperature 1.917°C lower than the classical approach, which in the total costs function, Φ, terms represents an improvement of 34.94%.

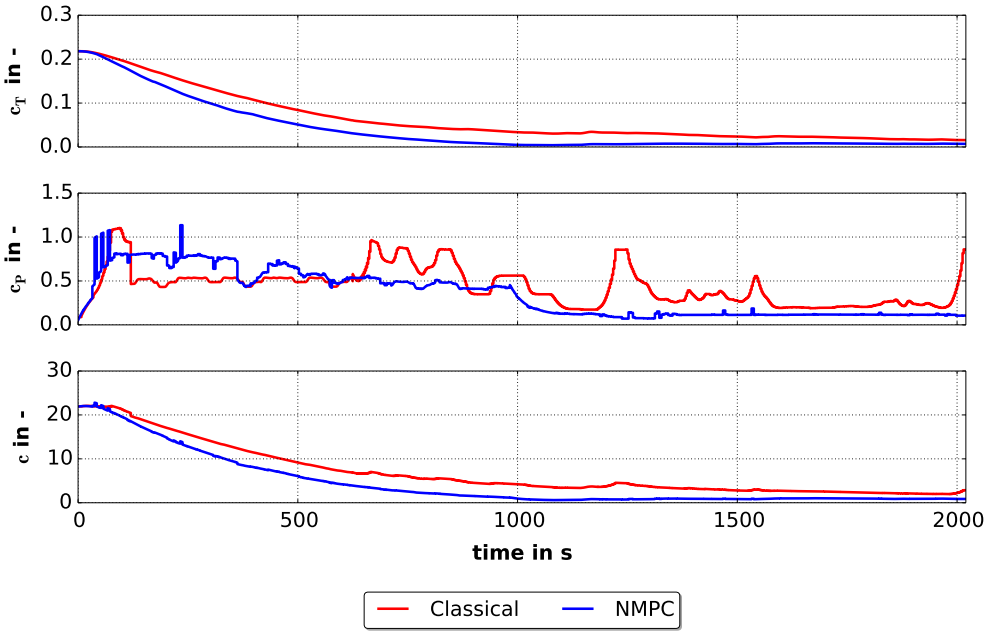


FIGURE 5.17: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the EMPA B driving cycle under hot conditions.

5.6.4 EMPA B under mild conditions

Figure 5.18 shows the results for the EMPA B under mild conditions.

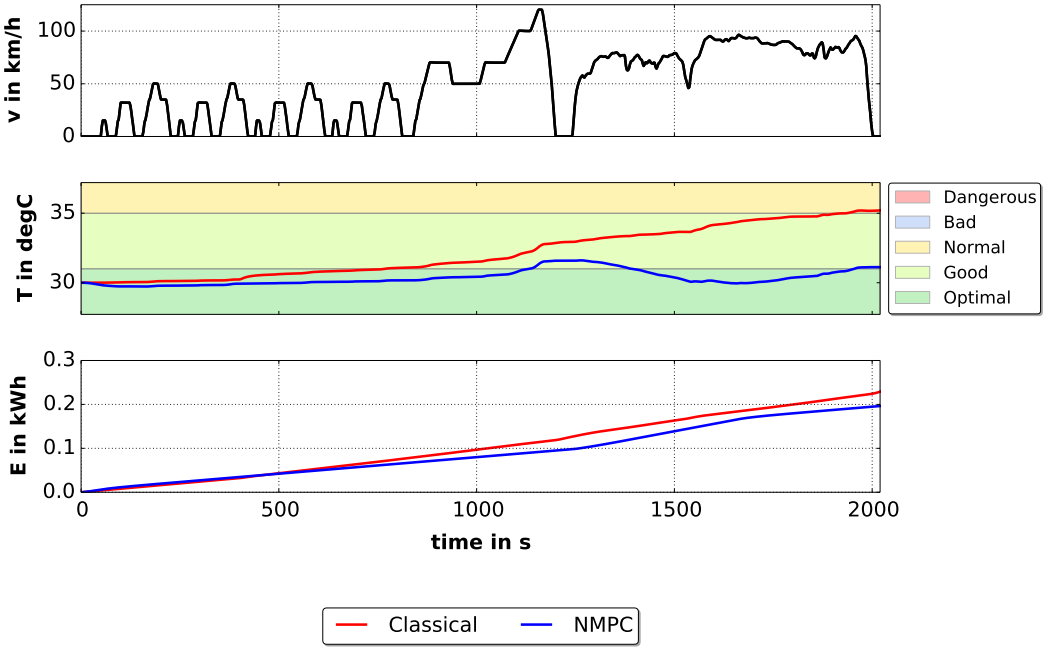


FIGURE 5.18: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the EMPA B driving cycle under mild conditions.

As it can be seen in the middle plot, the NMPC is able to keep the BAT temperature nearly over the whole time inside the "optimal" range, while the classical strategy even reaches the "normal" range, in yellow. At the same time, the NMPC consumes less electrical energy.

In figure 5.19 the controls that lead to the different TM can be seen. Taking a look at the classical approach, red lines in the plots in figure 5.19, two differentiate working points can be distinguished. During the first 400 s, since the initial BAT temperature is already in the "optimal" range, the classical approach starts with an energy saving configuration: fan off and pumps driven with minimal PWM. The cooler is bypassed, top left plot in figure 5.19, although the BAT temperature is slightly above 30°C figure 5.20, because the cooling condition of figure 5.6 is not fulfilled and thus there is not enough temperature difference between coolant and air to activate the cooler.
However, after 400 s the cooling condition is fulfilled and thus the one circuit mode together with the cooler, fan and pumps are used to cool down the BAT.

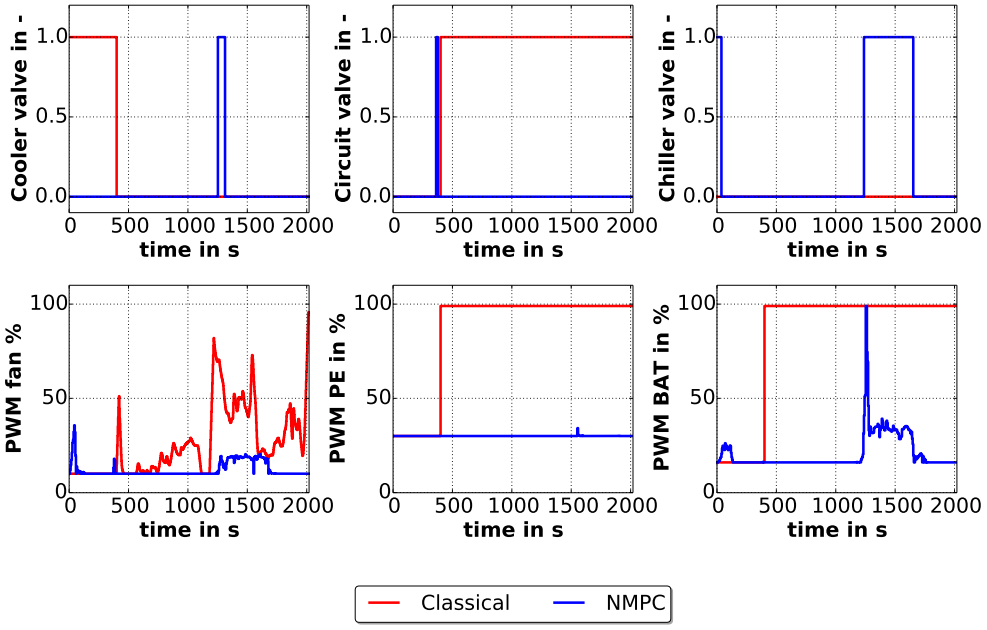


FIGURE 5.19: NMPC vs Classical TM controls for the EMPA B driving cycle under mild conditions.

On the opposite, the NMPC exploits the cooler potential from the beginning (Cooler valve = 0) and as the classical approach, minimizes the consumption of the pumps and fan. Nevertheless, after the highway part of the NEDC (last 800 s), the NMPC uses the chiller to cool down the BAT making less use of the fan. The reason for this fact is that the temperature difference between air and coolant is not appealing enough, contrary to the hot scenario case.

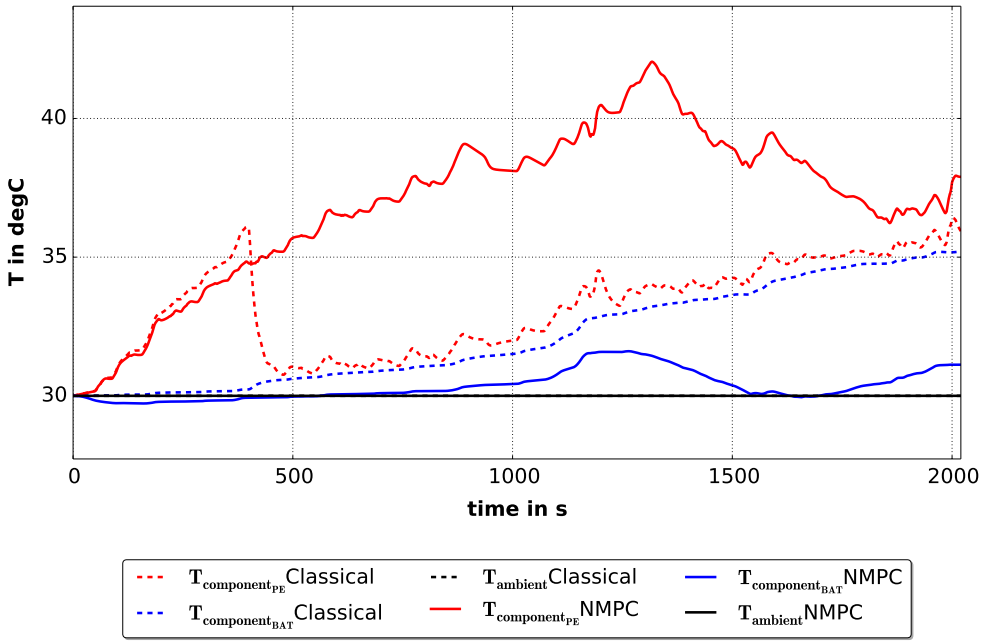


FIGURE 5.20: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the EMPA B driving cycle under mild conditions.

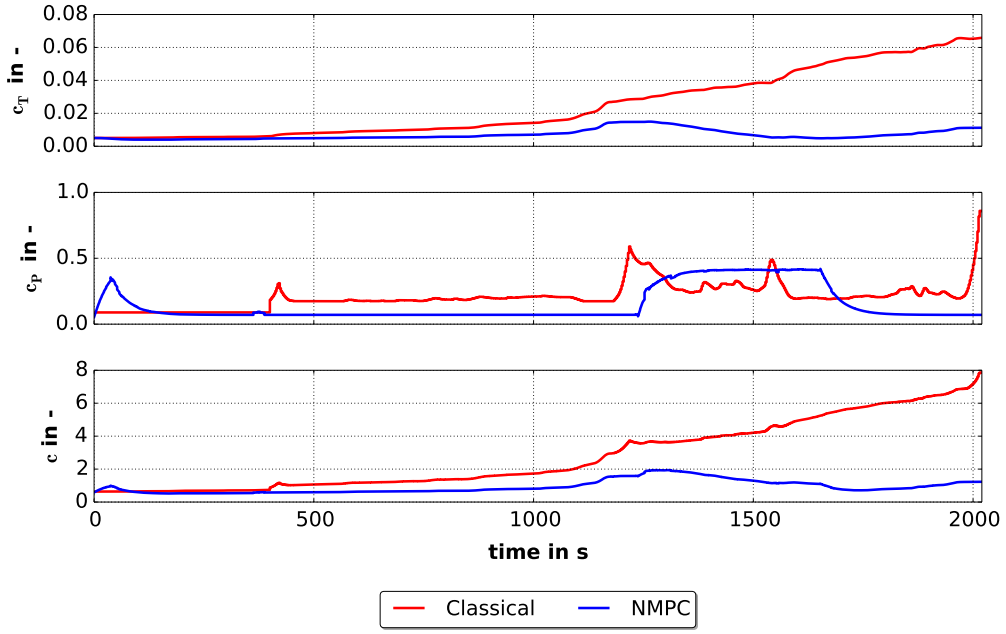


FIGURE 5.21: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the EMPA B driving cycle under hot conditions.

The different values for the circuit valve: two circuit mode in NMPC and one circuit mode in the classical strategy, top middle plot in figure 5.19, lead to different temperature levels

in the components. As shown in figure 5.20 the PE and BAT, dotted lines, present nearly the same temperature in the classical approach while in the NMPC they are far from each other.

In the top plot in figure 5.21 it is quite clear that the use of the fan from the second 400, starts penalizing moderately the classical strategy, while the clear deviation of both costs takes place when the fan PWM signal exceeds the 50% at time ≈1200 s. At the same time, the middle plot in figure 5.21 shows that the use of the chiller in the NMPC in the range 1200-1700 s, supposes less costs than the use of the fan in the classical strategy from second 480 till the end of the cycle. This fact leads to an electrical consumption reduction of 14.27%. Additionally, despite the lower investment in electrical consumption, the average BAT temperature in the NMPC is 1.799 °C better than in the classical control. Then the total costs function improvement represents 66.67%.

5.6.5 EMPA B under cold conditions

Figure 5.22 shows the results for the EMPA B cycle under cold conditions. The middle plot evidences that the temperature trajectories with the two strategies is very similar and the bottom plot that the same occurs with the electrical consumption.

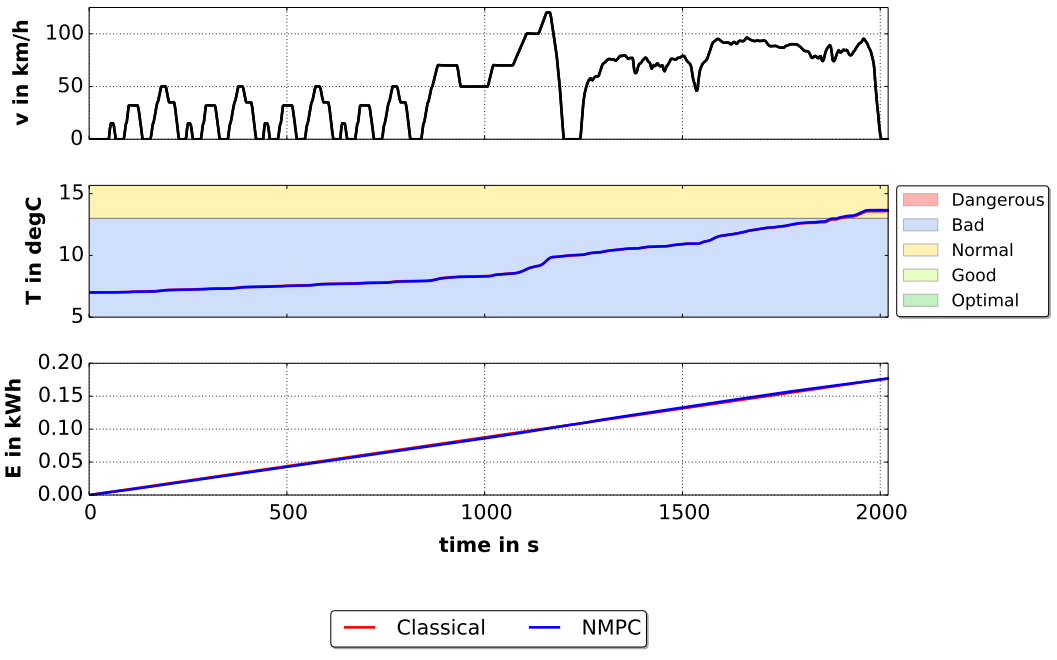


FIGURE 5.22: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the EMPA B driving cycle under cold conditions.

The strategy in both approaches, as it can be seen in the plots in figure 5.23, consists of coupling the PE and BAT circuits, bypassing the cooler and not using the chiller. Given that the initial temperature range is "bad", blue area in figure 5.22, with the one circuit and bypass modes it is possible to use the heat generated in the PE to warm up the BAT avoiding the heat transfer with the air which is colder than the coolant.

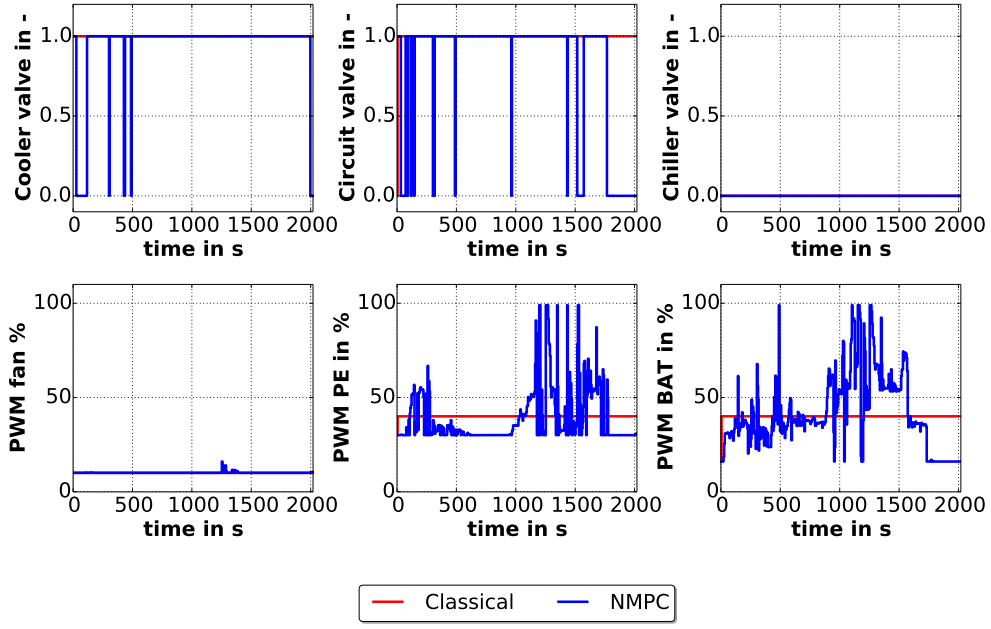


FIGURE 5.23: NMPC vs Classical TM controls for the EMPA B driving cycle under cold conditions.

The only difference is that NMPC uses, during short intervals, the valves positions that do not consume energy: Cooler valve $= 0$ and Circuit valve $= 0$ and moreover it constantly plays with the PWM signals of the pumps. These modes lead to the slightly different PE temperature trajectories, in the red continuous and dotted lines in figure 5.24.

As it can be seen in the middle plot of figure 5.25, the efforts of the NMPC to save energy during the first 100 s and the last 200 s, where the blue line is below the red one, are compensated with the higher power in the 1100-1600 s interval. Thus being the electrical energy reduction at the end of the cycle only 0.015% and supposing no success in the task of increasing the average BAT temperature. Given the larger weight of the temperature in the total costs function, the final result is no improvement.

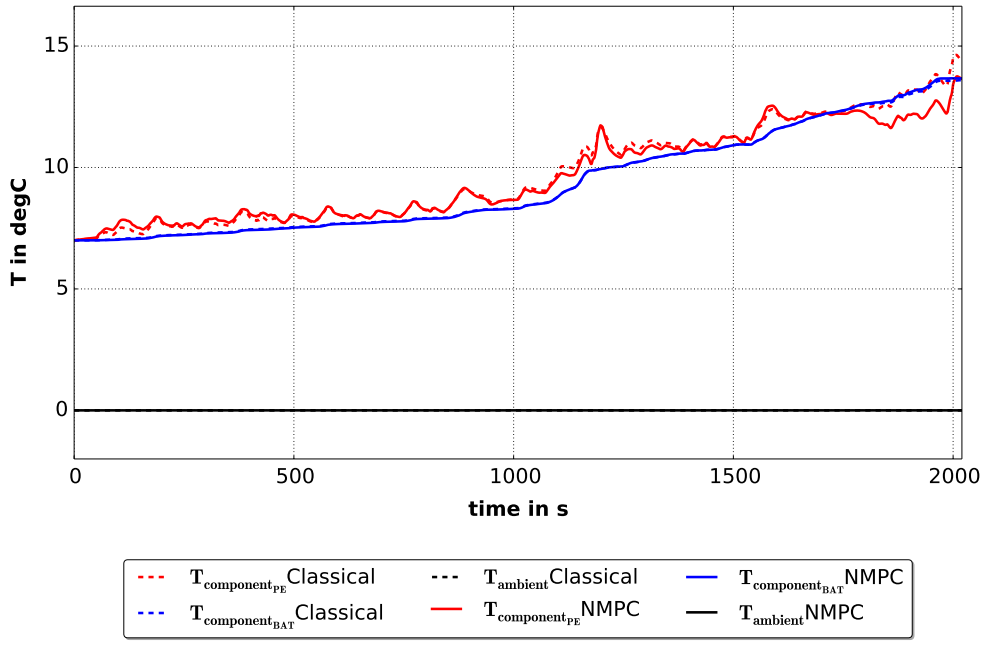


FIGURE 5.24: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the EMPA B driving cycle under cold conditions.

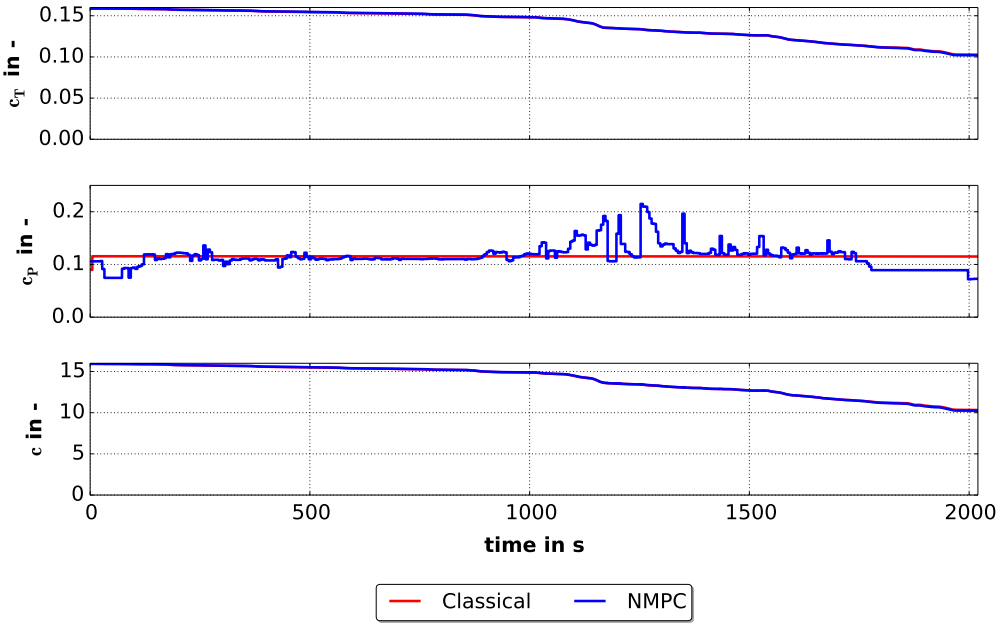


FIGURE 5.25: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the EMPA B driving cycle under cold conditions.

5.6.6 Constant cycle under hot conditions

Figure 5.26 shows the results for the constant cycle under hot conditions. The NMPC control strategy presents an improvement in the two goals: the electric consumption is reduced and the temperature trajectory reaches the "good" region, in contrast to the classical control.

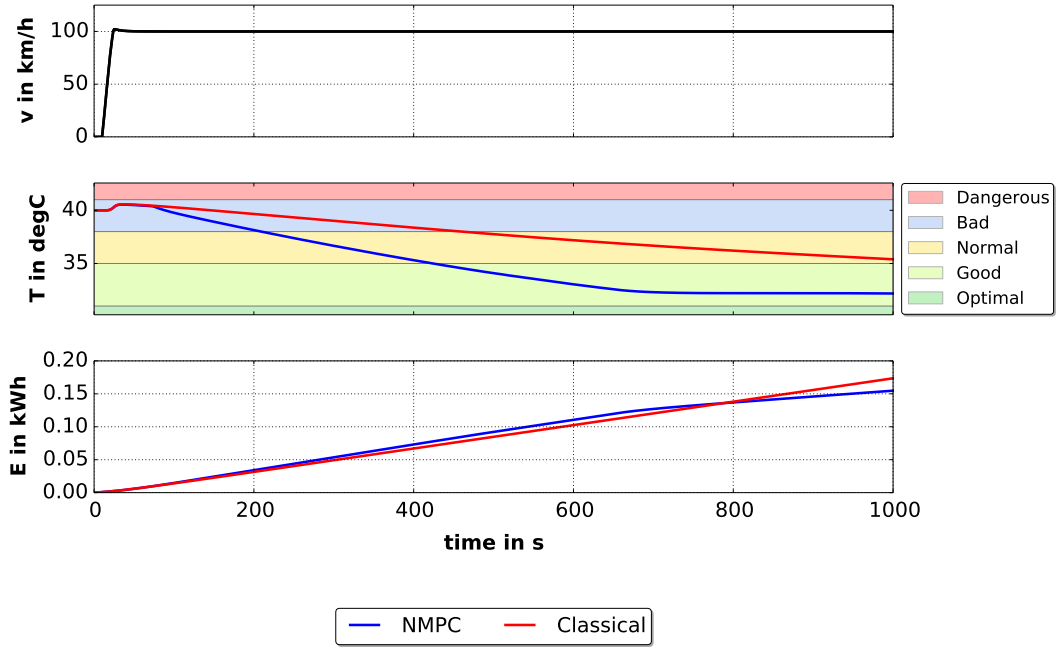


FIGURE 5.26: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the constant driving cycle under hot conditions.

Due to the static profile of the constant cycle and the large difference between the BAT temperature and the ambient, the NMPC controls, shown in the blue lines of figure 5.27 are less variable than in the previous discussed scenarios. The most of the time, the NMPC relies on the configuration: one circuit mode, chiller active and cooler active. The PWM signal of the fan, bottom left plot in figure 5.27, is used moderately given that the air mass flow is high enough at 100 km/h. The classic strategy, on the opposite, does not use the cooler and the chiller at the same time. For this reason the NMPC can save more than 300 s of chiller operation, as it can be seen in the top right plot in figure 5.27.

Again the one circuit mode in the NMPC and the two circuit mode in the classical strategies, involve the closed trajectories of the BAT and the PE, red and blue solid lines in figure 5.28 and the clear difference shown in the blue and red dotted lines.

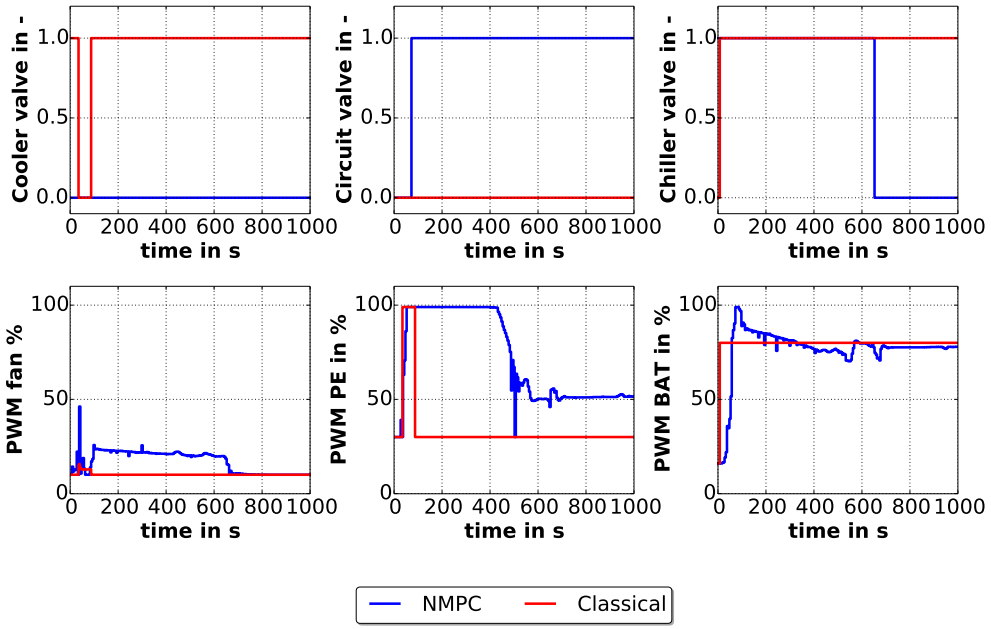


FIGURE 5.27: NMPC vs Classical TM controls for the constant driving cycle under hot conditions.

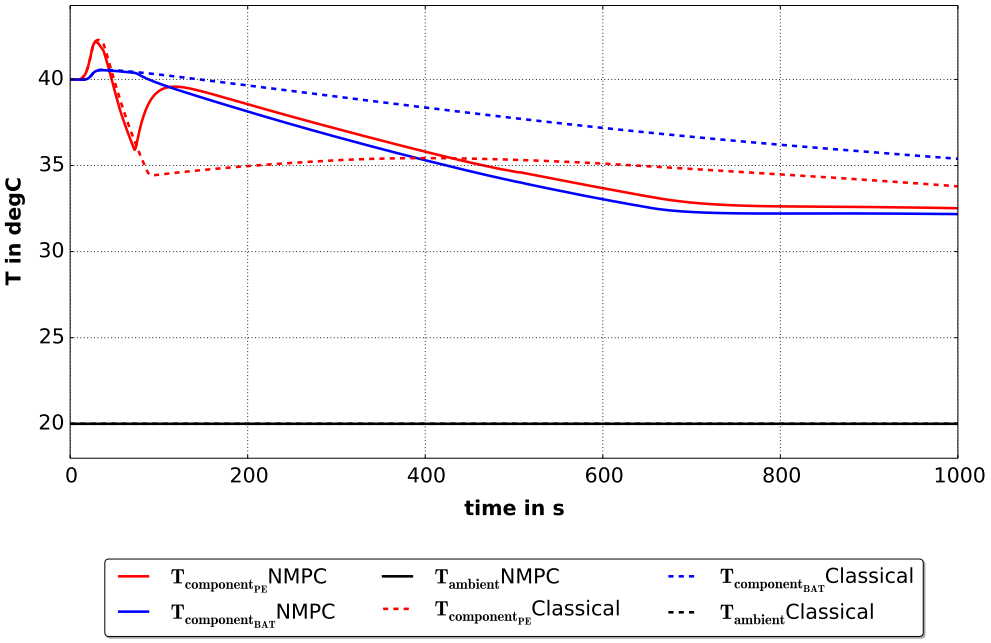


FIGURE 5.28: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the constant driving cycle under hot conditions.

The effect of using the cooler and the chiller at the same time can be seen in the faster negative slope of the blue line in the temperature cost term c_T in the top plot in figure 5.29.

The middle plot in figure 5.29 shows that the power cost term c_P is higher in the NMPC during the first 650 s as it uses higher PWM signals in the pumps, bottom middle and right plots in figure 5.27. Nevertheless, in the last 300 s the NMPC strategy does not use the chiller, which leads to the considerable decrease in the power costs in the middle plot of figure 5.29.

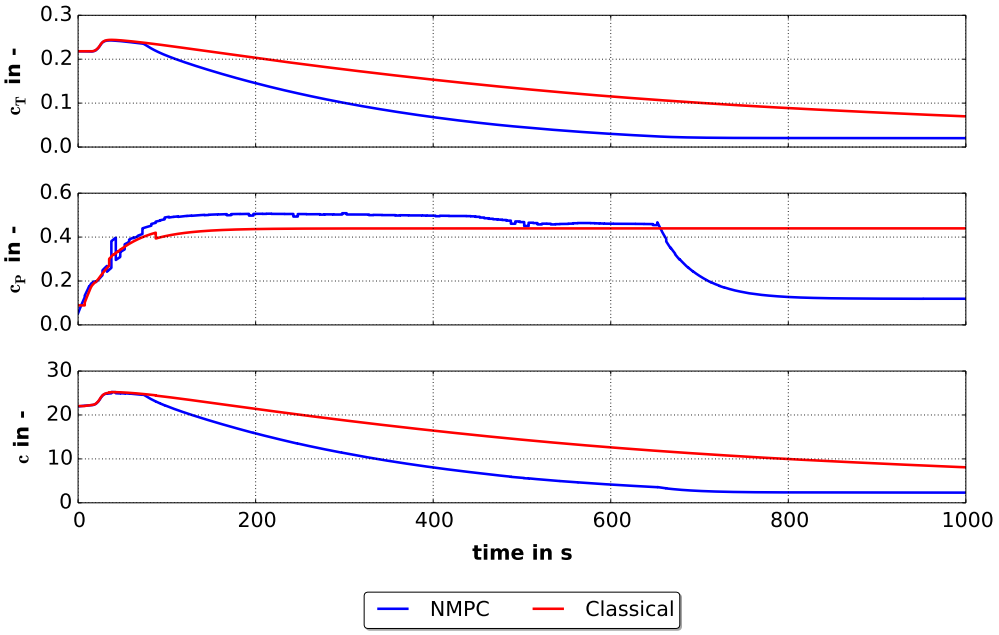


FIGURE 5.29: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the constant driving cycle under hot conditions

Finally, the lower total costs c shown in the bottom plot of figure 5.29 can be translated in a reduction of the electrical consumption of 10.84% together with a decrease in the average BAT temperature of 2.876 °C which yields a total costs reduction of 42.91%.

5.6.7 Constant cycle under mild conditions

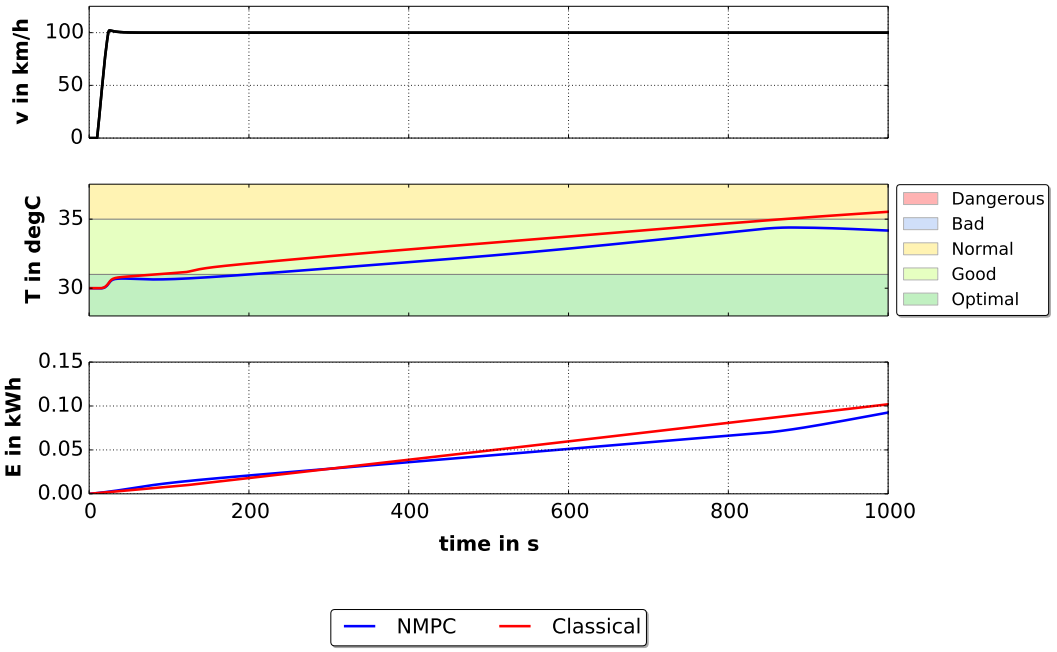


FIGURE 5.30: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the constant driving cycle under mild conditions.

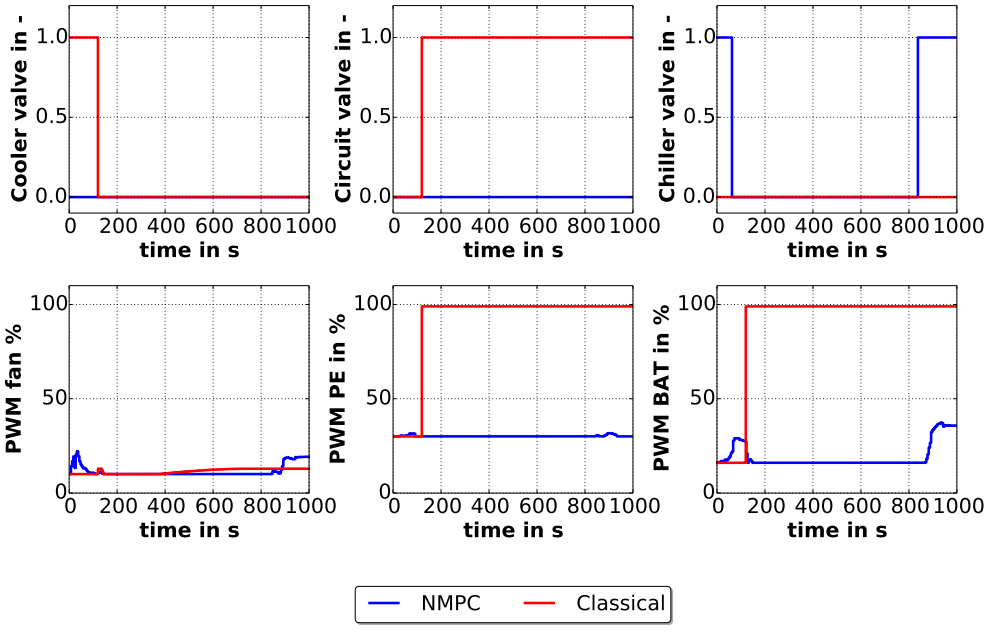


FIGURE 5.31: NMPC vs Classical TM controls for the constant driving cycle under mild conditions.

The classical approach starts with the two circuit mode and cooler bypassed, top left and middle plots in figure 5.31, and as soon as the cooling condition is fulfilled, the cooler is activated and the circuits are coupled with the pumps rotating at full speed, bottom middle and right plots in figure 5.31. The NMPC control strategy on the contrary is based on the two circuit mode, being the cooler active and the pumps at minimum power operation. The chiller is used at the beginning and the last 180 s to cool down the BAT. Again the different circuit operation modes of both strategies lead to considerable differences in the components temperatures, as it can be seen in figure 5.32.

The effect of the chiller in the temperature can be seen in the blue line in the top plot of figure 5.33, where at the beginning and the end of the cycle, when the chiller is active, the slope of the costs associated to the BAT temperature decreases significantly. The prize to pay for this refrigeration can be seen in the power costs, middle plot in figure 5.33, where the blue line is above the red one in the mentioned intervals. The rest of the cycle, thanks to the non consuming valves positions and the moderate PWM signals, the NMPC control strategy can save energy. All in all, the NMPC achieves a reduction in the electrical consumption of 9.190% and a 0.764°C lower average BAT temperature, which implies an improvement in the total costs reduction of 26.7%.

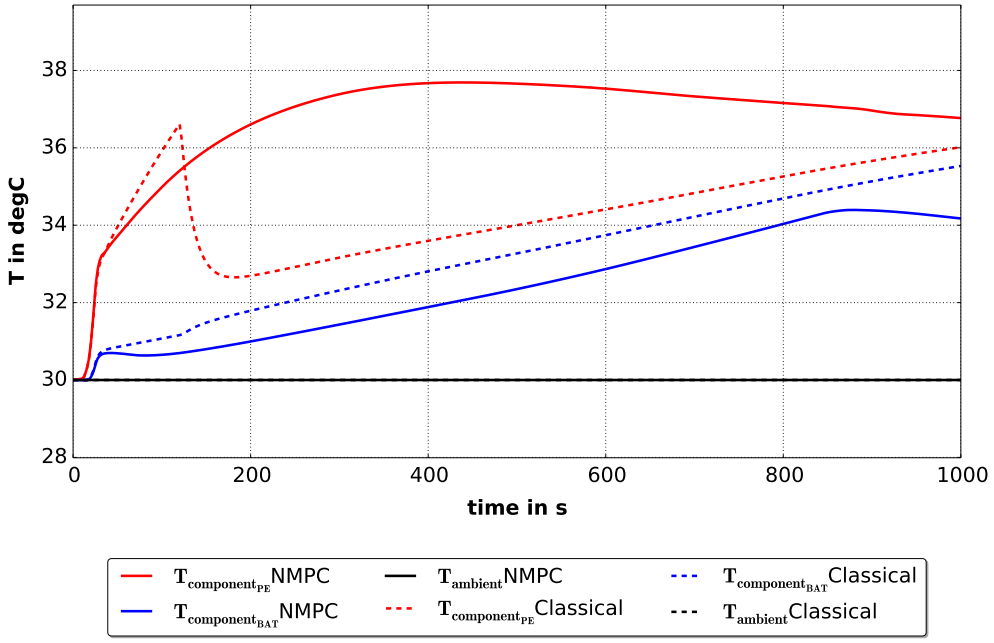


FIGURE 5.32: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the constant driving cycle under mild conditions

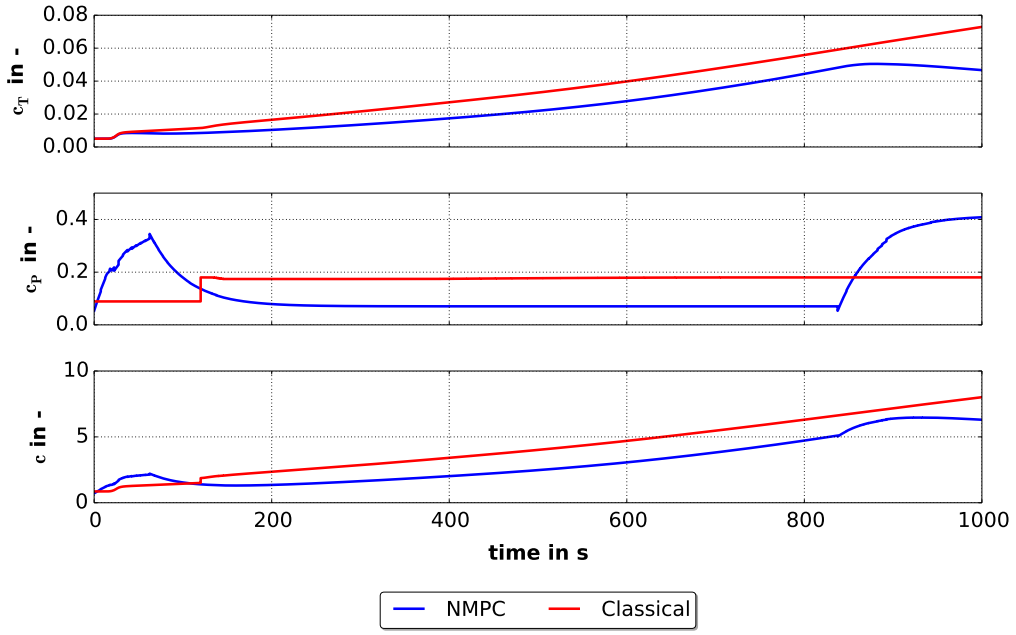


FIGURE 5.33: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the constant driving cycle under mild conditions.

5.6.8 Constant cycle under cold conditions

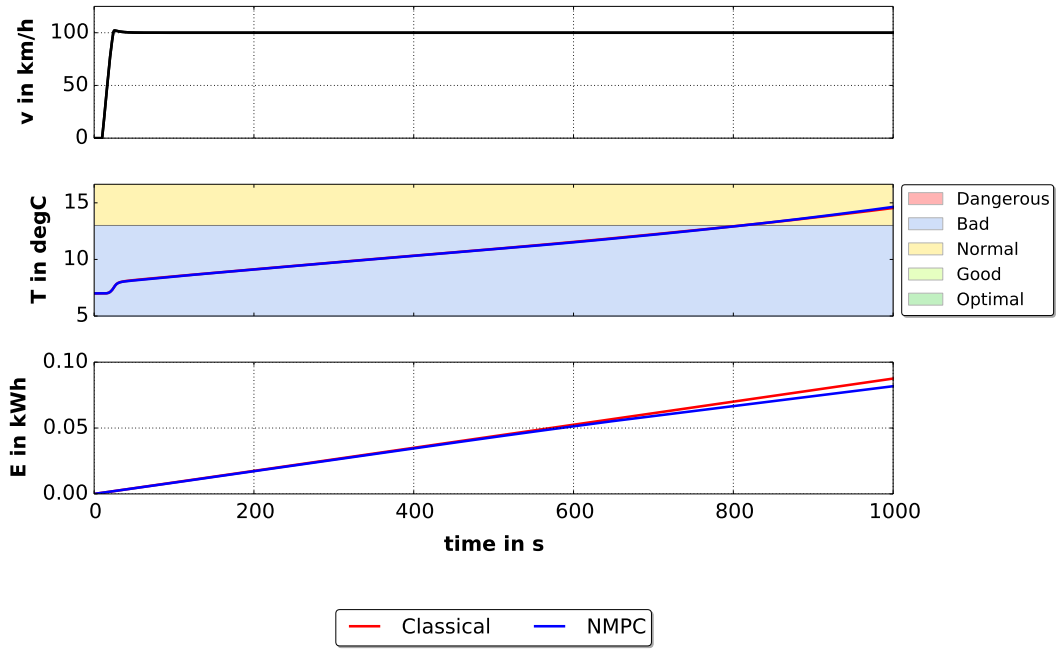


FIGURE 5.34: NMPC vs Classical TM goals: BAT temperature trajectory and electrical consumption of the actuators for the constant driving cycle under cold conditions.

In the case of the constant cycle under cold conditions, the improvement of the NMPC TM is only partial, since the BAT temperature does not show a faster heating and only the electrical consumption is reduced, as shown in the middle and bottom plots of figure 5.34.

Both strategies try to heat the BAT with the PE generated heat using the one circuit mode and avoiding the ambient air through the cooler bypass, since the air is colder than the coolant. This can be seen in the top left and middle plots in figure 5.35. The only difference of both strategies is that NMPC tries to save some electrical energy with the valves position in the last 300 s, cooler valve $= 0$ and circuit valve $= 0$, and with the pumps, as it can be seen in the bottom middle and right plots of figure 5.35. This can also be seen in the last 500 s of the middle plot in figure 5.37, where the NMPC costs associated to the electrical power are lower than the classical. In energy terms, the final decrease is 6.63%. The improvement in the two goals leads to 0.2% lower total costs. As a consequence of the cooler active, one circuit mode to save energy, a remarkable decrease in the PE temperature is observed during the last 300 s, red solid line of figure 5.36.

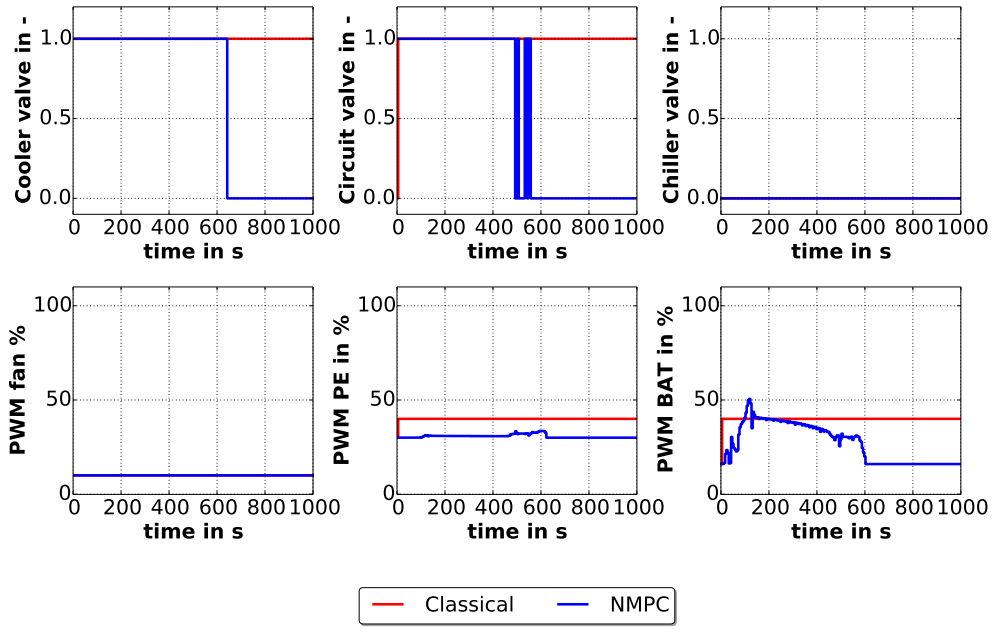


FIGURE 5.35: NMPC vs Classical TM controls for the constant driving cycle under cold conditions.

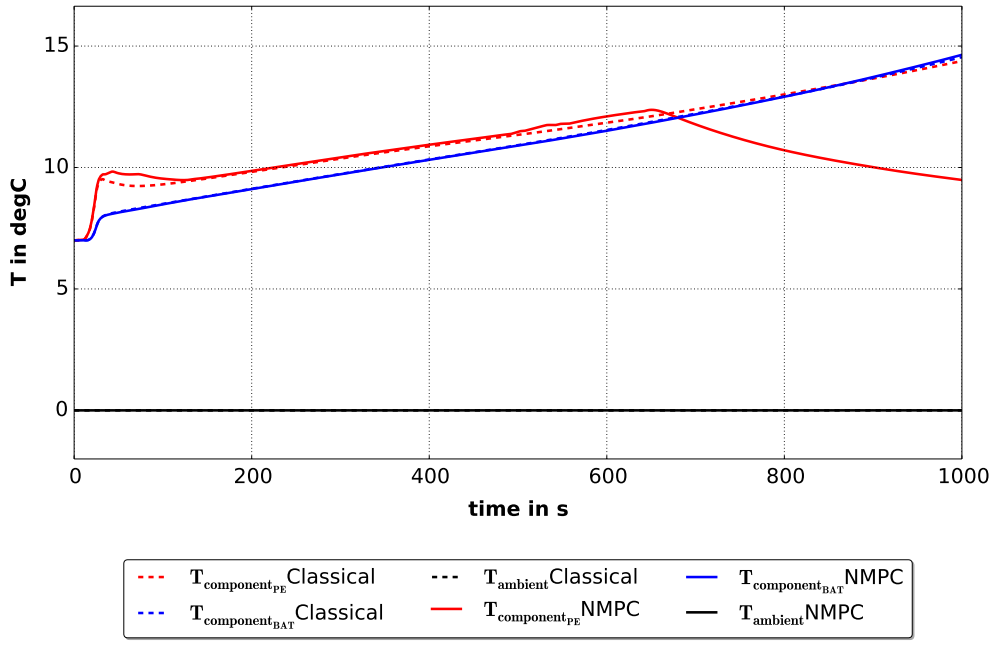


FIGURE 5.36: NMPC vs Classical TM main temperatures: BAT, PE and ambient for the constant driving cycle under cold conditions.

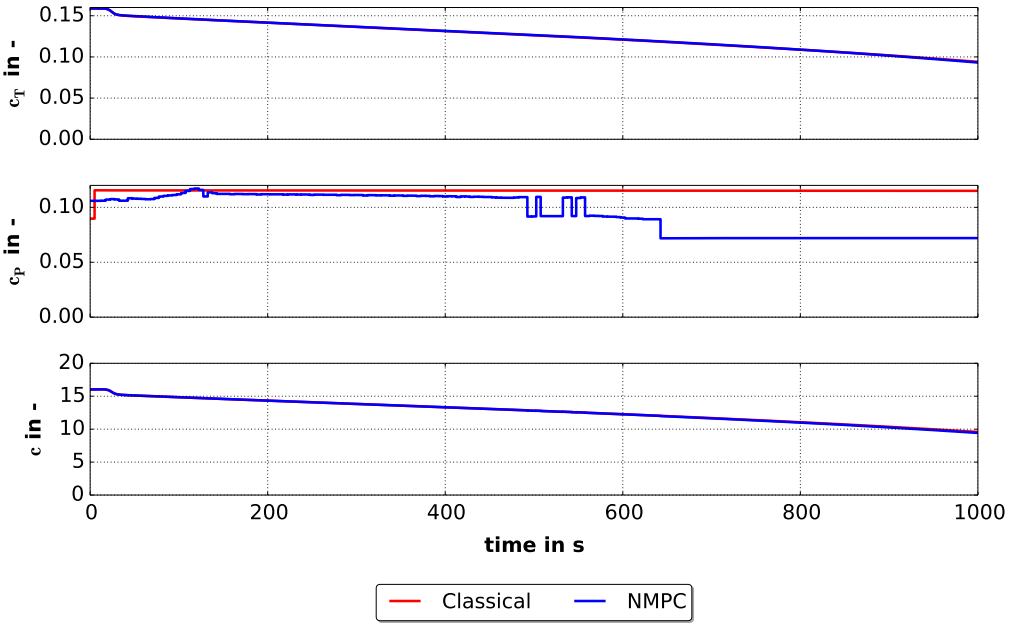


FIGURE 5.37: NMPC vs Classical TM costs associated to the BAT temperature and the electrical consumption of the actuators for the constant driving cycle under cold conditions.

5.7 Simulation results discussion

Table 5.2 summarizes the simulation results presented in the previous sections using the following variables:

$$\Delta\mu_{T_{BAT}} = \mu_{T_{BAT_{NMPC}}} - \mu_{T_{BAT_{Classical}}}, \tag{5.1a}$$

$$\Delta E_{LT2}(t_{end}) = \frac{E_{LT2_{NMPC}}(t_{end}) - E_{LT2_{Classical}}(t_{end})}{E_{LT2_{Classical}}(t_{end})}\,100 \tag{5.1b}$$

$$\Delta\Phi = \frac{\Phi_{NMPC}(t_{end}) - \Phi_{Classical}(t_{end})}{\Phi_{Classical}(t_{end})}\,100 \tag{5.1c}$$

where (5.1a) stands for the difference of the average μ BAT temperatures in the cycle and (5.1b) is for the difference in the final electrical consumption of the circuit in relative terms. For the hot and mild scenarios, a decrease in the average temperature is a success, represented in green in the table, while in the cold scenario it is a failure, in red in the table. The electrical consumption and the total costs reduction imply an improvement and hence are colored green when they are negative and red when positive.

The first thing that must be highlighted is that excluding the electrical consumption in the cold EMPA B scenario, the rest of goals are improved with the NMPC control strategy. While the best numbers belong to the hot scenarios, the cold ones do not introduce a real improvement. The reason for this fact is that the LT2 circuit presents several heat sinks for cooling the components but no heat sources for warming them up. Hence, the only possible strategy is to use the heat losses dissipated in the PE, which has a lower thermal mass, to achieve a faster heating of the BAT.
For the rest of scenarios, transient and static driving cycles and mild and hot conditions, the NMPC shows a significant improvement in the two TM goals, being the temperature improve around 1 and 3 °C and the energy consumption around 9 and 14%.

Additionally, it must noted that no violation of the temperatures constraints was observed in any of the studied driving cycles. This way, the PE was always kept under 60°C.

TABLE 5.2: TM simulation results

Conditions / Cycle	Hot			Mild			Cold		
	$\Delta\mu_{T_{BAT}}$	ΔE_{LT2}	$\Delta\Phi$	$\Delta\mu_{T_{BAT}}$	ΔE_{LT2}	$\Delta\Phi$	$\Delta\mu_{T_{BAT}}$	ΔE_{LT2}	$\Delta\Phi$
EMPA B	-1.917 °C	-14.12 %	-34.94 %	-1.799 °C	-14.27 %	-66.67 %	-0.001 °C	-0.015 %	+ 0.003 %
Constant	-2.876 °C	-10.84 %	-42.91 %	-0.764 °C	-9.19 %	-26.7 %	+0.007 °C	-6.630 %	-0.2 %

Chapter 6

Nonlinear Model Predictive Control: Vehicle Results

"It's not an experiment if you know it's going to work."

— Jeff Bezos

CONTENTS:

This chapter deals with the NMPC approach validation in a PHEV prototype. After describing the vehicle implementation required for a successful communication between controller and controlled plant, the results obtained in different test drives will be presented.

6.1 Vehicle implementation for Real-time NMPC

Although the NMPC validation through simulations is a necessary step and can be used to set several key aspects for an appropriate performance, the final goal of this study was to achieve a real-time implementation of the NMPC in a real vehicle. More precisely, as shown in figure 6.1, the goal was to enable a successful real-time communication between MUSCOD-II running on a laptop and the vehicle. In contrast to the NMPC method discussed in Chapter 5, here the exchange of the states x and controls u was done asynchronously. In other words, as soon as a new computation of the below discussed real-time method is available, the data exchange takes place.

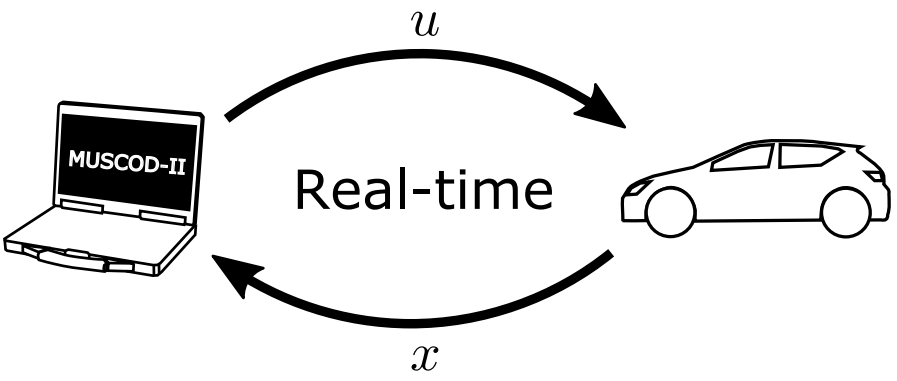


FIGURE 6.1: Nonlinear Model Predictive Control loop.

FIGURE 6.2: Test vehicle instrumentation: Coolant flow meters and thermocouples in the motor compartment (left) and air thermocouples on the roof (top right) and in front of the cooler (bottom right).

6.1.1 Controlled plant states x data acquisition

The mechanical states of the controlled plant v, M, n, P_{car} and other relevant general information were acquired reading the existing Controller Area Network (CAN) buses in the vehicle. The rest of the controlled plant states x (4.10) were read by means of an extra CAN bus specially built for measuring the output of 17 sensors equipped in the Golf GTE prototype. In total 17 thermocouples of type K with accuracy of $\pm 1°C$ were used to measure the coolant temperature in 15 points of the circuit, the air in front of the cooler and on the roof of the vehicle. In addition, three turbine flow meters with a linearity of 0.1% were used to measure the coolant volume flow rate. Figure 6.2 shows the different sensors equipped in the vehicle. The software used for data acquisition was INCA® from ETAS GmbH.

6.1.2 Control signals u access

The cleanest option for overtaking the controls of the LT2 cooling circuit actuators was to bypass the existing configuration in the vehicle via a RP module. The other alternative would have been to cut the physical connections to the actuators. Nevertheless, this would have required: 1) to build the physical signals, of PWM type among others, according to the MUSCOD-II calculated controls 2) to disable the diagnosis code related to the manipulated actuators (seen "in air" from the Electronic Control Unit (ECU)) and finally 3) to make the physical connections. To avoid possible failures and save time, the bypass option using the commercial RP module ES910 together with the software ASCET® and EHOOKs® from ETAS GmbH was chosen. This solution is further discussed in the next section.

6.1.3 Harware implementation in the vehicle

With the aim of being able to compare the classical control with the NMPC in successive driving tests, the design in figure 6.3 was implemented. With this implementation, it can be easily switch between the two following operation modes:

- **NMPC:** MUSCOD-II runs in the Laptop held by the co-pilot, being connected to the RP module through an Ethernet connection. The control signals are sent by means of the Universal Serial Bus (USB) connected Controller Area Network (CAN) card to the RP module. The ECU no. 1 is equipped with an Emulator test probe (ETK) that allows that the control signals arriving to the ECU via the RP ETK connection, are taken instead of the original code in it. This way the original physical electric connections to the actuators in the cooling circuit can be kept. Furthermore, the states of the controlled plant, output signals of the temperature sensors installed in the cooling circuit and

other signals running in the can buses of the vehicle are sent to MUSCOD-II through the RP module.

Since the chiller valve is physically stimulated from another ECU, ECU no. 2, that is not equipped with ETK, a CAN logger is needed (top right corner of figure 6.3). The CAN logger performs a gateway that splits the CAN bus containing the original command for this valve. This way the $Valve_{CHILLER}$ calculated in MUSCOD-II can be used instead of the original vehicle demand.

- **Classical:** The RP deactivates the bypass of the ECU no. 1 and the CAN logger sends the signal arriving from the original CAN bus to the ECU no. 2. In this mode, the original control signals of the vehicles for the cooling circuit and AC circuit are taken. These control signals are set to constant values output by a finite-state machine, similar to the one used in Chapter 5, with four possible states: heating, temperature maintaining, mild cooling and maximal cooling. The conditions for changing from one state to another depend on the current BAT temperature and some sensors describing the availability of the heat exchangers to dissipate the heat.

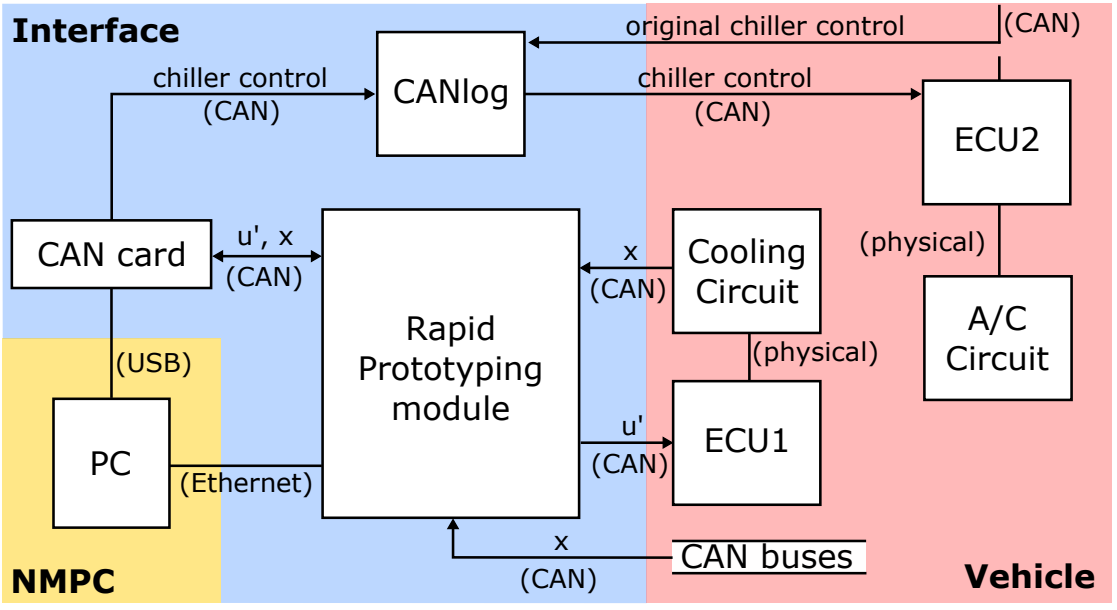


FIGURE 6.3: Hardware implementation for the cooling circuit control manipulation.

6.2 Test drives

6.2.1 Repeatable driving cycles

In order to compare the TM resulting from the NMPC and the classical control strategies, two driving cycles were designed according to the following criteria:

- The cycle should be performed in an open-accessible street.
- The cycle should be repeatable.
- The cycle should imply a considerable thermal load in the electric components.

Although the validation in a test bench would have been more suitable from the repeatability point of view, for time and costs reasons the open-accessible street option was preferred. Compared to the test bench alternative, the open-accessible street offered a more comfortable margin in case of mishaps in the initial startup procedure of the complex hardware and software implementation.

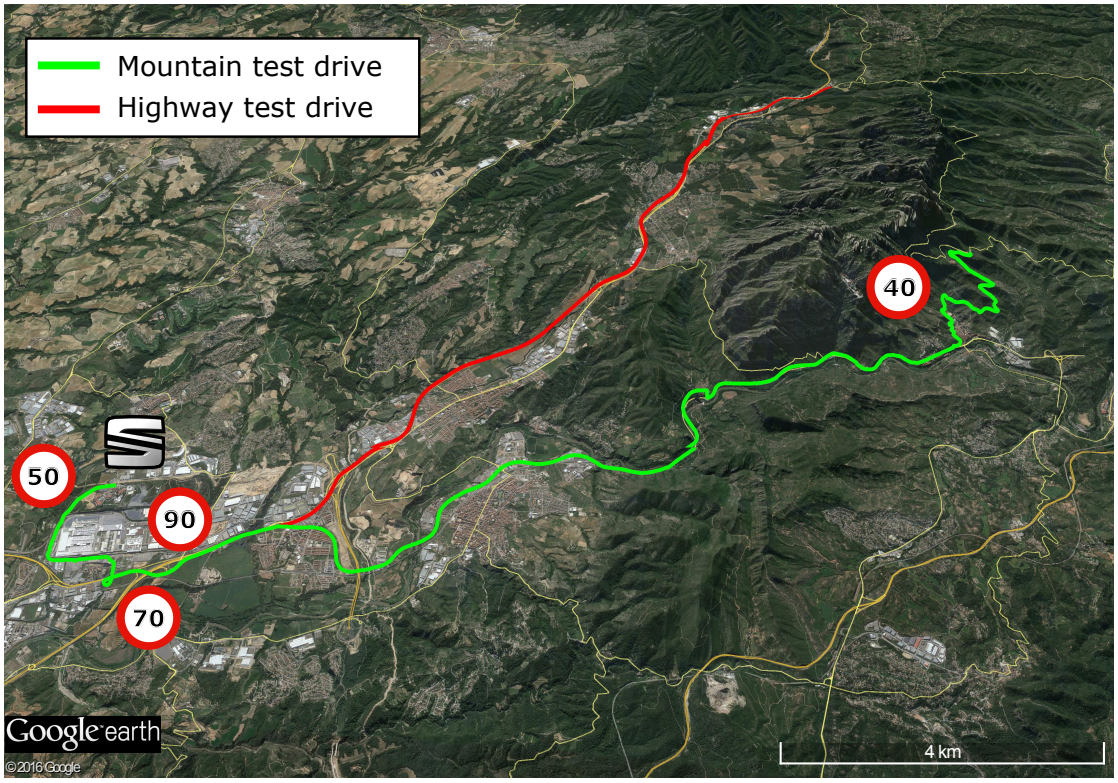


FIGURE 6.4: Test drives designed for NMPC validation.

On the other hand, repeatability is difficult to achieve circulating in open-accessible with the presence of other vehicles which can be seen as obstacles in the road that prevent the

vehicle from following the repeatable cycle forcing the driver to accelerate or break abruptly. To minimize the traffic disturbances, the ACC that is available in the car was used to drive the car in the different sections of the roads. This way the same cycle can be driven twice, once with the NMPC and once without, being the accelerations and decelerations performed with the ACC very smooth, in contrast to the driver natural reaction.

Figure 6.4 shows the two routes defined for the experiments which draw from the SEAT Technical Centre, Martorell-Barcelona(Spain). The speed limitations shown in this figure represent the speeds used for the ACC in each section of the road.

One way of achieving high thermal loads in the components is to perform a driving cycle which implies heavy mechanical loads, because the only heat generated in the LT2 cooling circuit is due to the *Joule effect* in the components. This is the case of the highway test drive, red route in figure 6.4, where the high mechanical load is achieved with the constant speed of 90 km/h and the increasing altitude of the road, as it can be seen in the top and bottom plots of figure 6.5, respectively. The total duration of this cycle is 21 km and it consists of a short section in a urban road, yellow area, followed by the mentioned highway part.

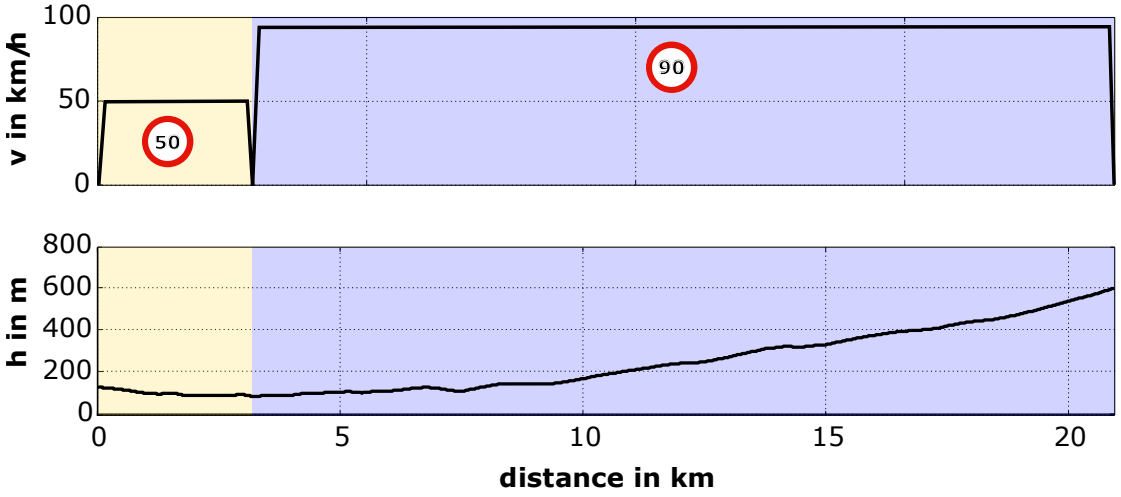


FIGURE 6.5: Highway test drive speed and altitude profiles.

Another key factor for achieving a challenging TM scenario is the total cycle duration, since the more time the components are working, the more heat is generated. The green route in figure 6.4, called mountain test drive, was designed to perform a long trip of 39 km under high mechanical loads. As it can be seen in the top plot of figure 6.6 this driving cycle is divided in a short urban section, yellow area, an interurban road, blue area and a road with considerable slope section, red and white areas.

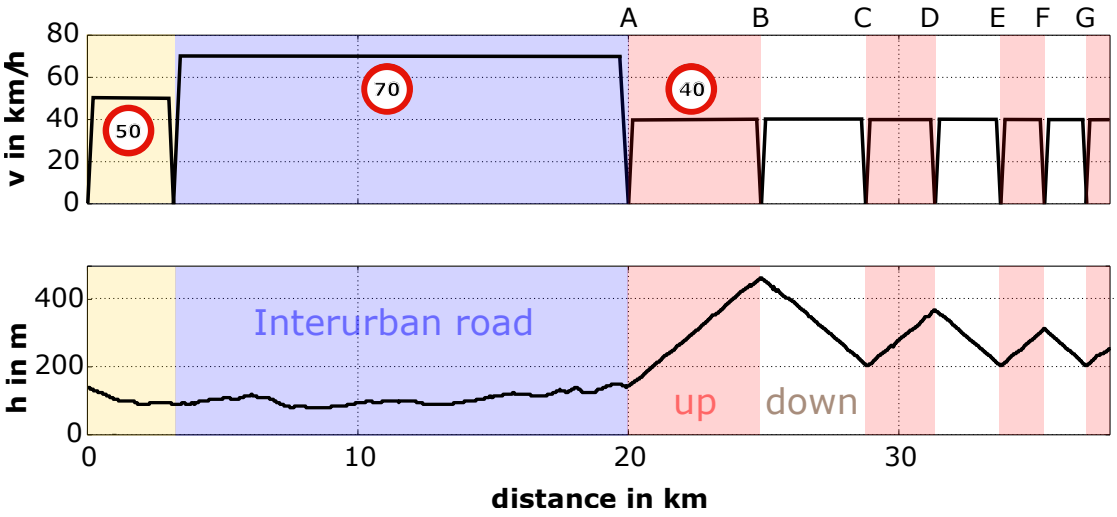


FIGURE 6.6: Mountain test drive speed and altitude profiles.

To achieve a long trip, several turning points, called A-G, were defined in the last part of the route. They are shown in the top plot of figure 6.6 and in the zoom of the road in figure 6.7.

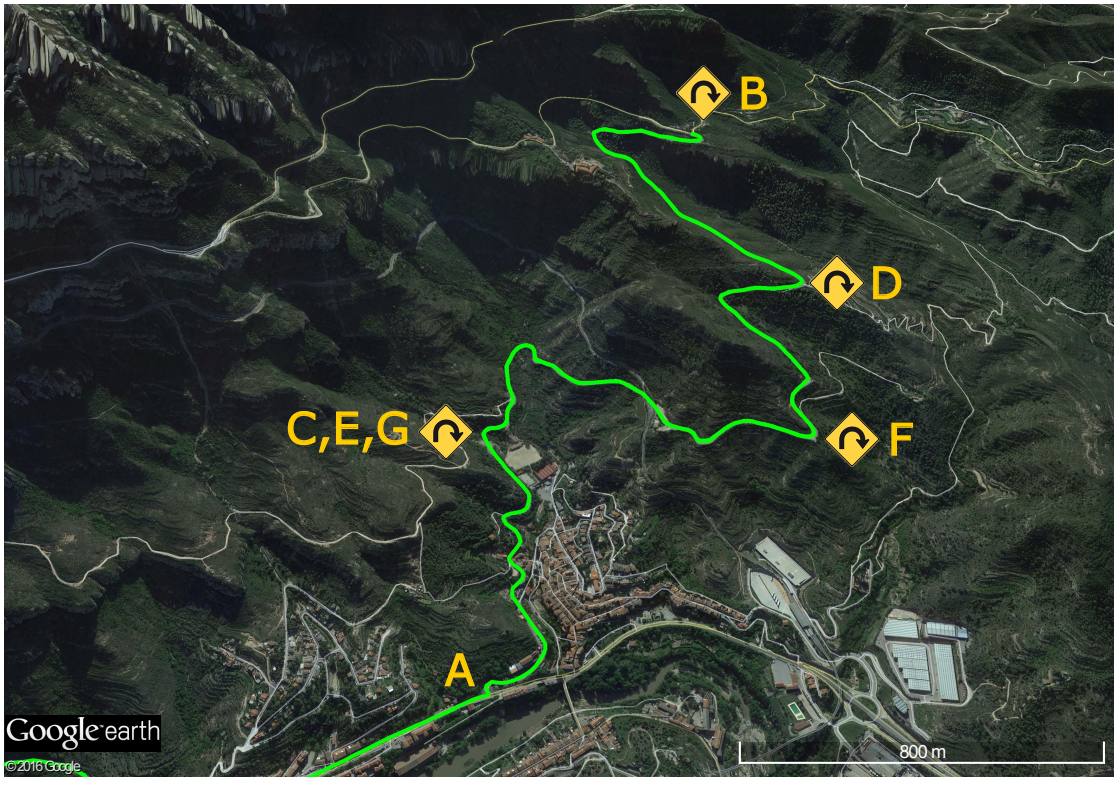


FIGURE 6.7: Zoom in the strategic turning points in the Mountain test drive.

The idea behind the definition of these strategic kilometric points was to use them as turning marks to perform several up- and downhills in the same road. Thus the vehicle faces the slope for the first time at A and drives till the highest point B is reached, where the vehicle turns over and starts the descent to the initial kilometric point C. Again, the vehicle turns

over and drives to the next turning point D, lower than B and so on till some meters after the last turn over in G, the BAT is fully discharged and the pure electric mode is no longer available.

With this driving strategy the BAT works supplying and receiving current and therefore generating heat during a long time in spite of the limited autonomy of the vehicle (50 km based on the NEDC which compared to the mountain test drive is a low load cycle).

6.2.2 Ambient and initial conditions

To consider that two measurements are comparable, the following requirements for the driving conditions were established:

- The initial state of the BAT had to be fully charged.
- The trip had to be performed in working days at similar hours from the traffic point of view.
- The wind conditions had to be similar.
- No stop apart from the turning points of the mountain test drive are possible.
- The maximal initial temperature $T_{component_{BAT}}$ difference between the two measures should not exceed 0.5 K in absolute terms.
- The maximal initial ambient temperature $T_{ambient}$ difference between the two measures should not exceed 2.5 K in absolute terms.

Additionally, to achieve always a similar electrical power demand to the BAT, all the tests were driven with the car being under the same conditions: auxiliary consumers like Heating, Ventilating, and Air Conditioning (HVAC) were turned off, as well as lights, radio and other electrical gadgets. Windows were opened to the same level and the weight of the car was held the same.
Several test drives were performed along this research. Table 6.1 shows the ones among them that finally resulted comparable:

TABLE 6.1: Thermal conditions for the NMPC validation test drives

Conditions / Cycle	Initial temperatures $T_{component_{BAT}}$	$T_{ambient}$
Mountain test drive hot	≈31°C	≈24°C
Mountain test drive mild	≈22°C	≈16°C
Highway test drive cold	≈14°C	≈14°C

Furthermore, it must be added that the cost function parameters $c_T = 100$ and $c_P = 2.5$ were taken for all the NMPC test drives.

6.3 Vehicle results

6.3.1 Mountain test drive under hot conditions

In this section, the results for the mountain test drive under hot conditions will be discussed. As it can be seen in figure 6.8, the NMPC and the classical control test drives, in solid and dotted lines respectively, draw from comparable thermal conditions: the BAT temperatures, blue lines, differ only 0.5 K and the ambient temperature, black lines, are practically the same.

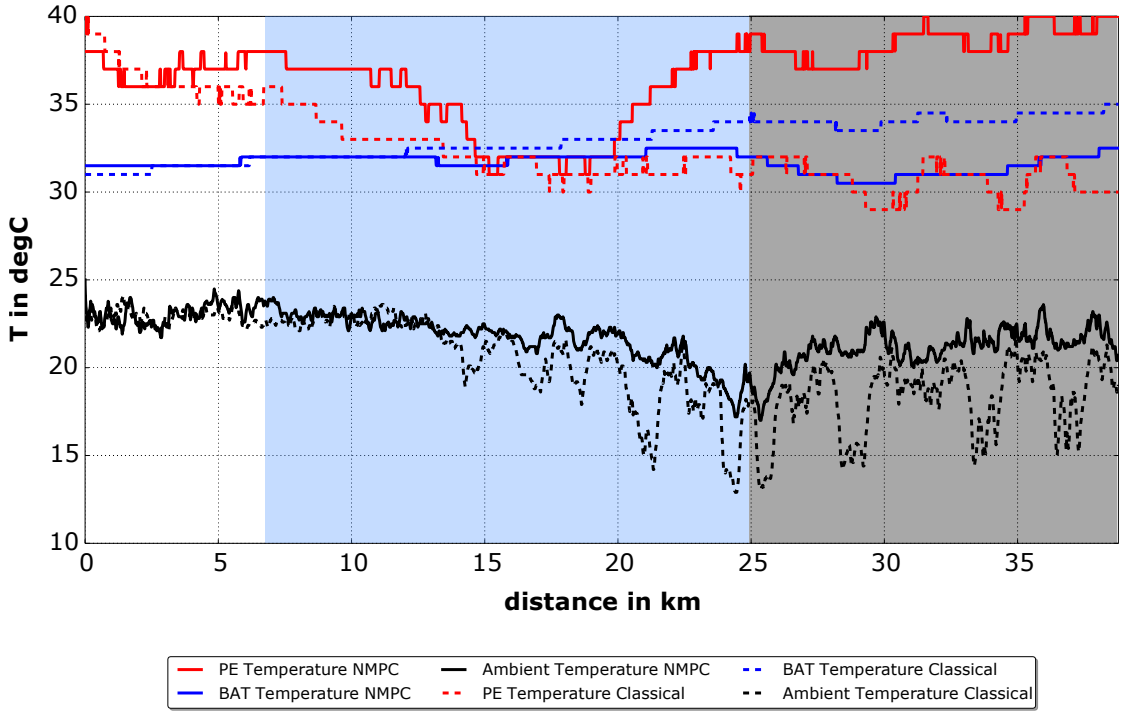


FIGURE 6.8: NMPC vs Classical TM temperatures in the mountain test drive under hot conditions.

Figure 6.8 also indicates that the start situation in this scenario is that the two components, BAT in blue and PE in red, are considerably hotter than the ambient air. Notice that the components temperatures show a step-wise trajectory. This is due to the fact that they represent the average of several sensors outputs calculated by the vehicle ECUs. These sensors and their average calculation are part of the series vehicle and in our experiments they were read in the appropriate CAN bus to be treated as the real temperature of the component.

Besides of drawing from a similar temperature state, both test drives performed the same speed profile with only small discrepancies due to punctual traffic in the road as it can be seen in the top plot of figure 6.9. Due to the fact that these small deviations in speed suppose different times for completing the entire route, from now on, it is more appropriate for the comparison of both controls to use the distance in kilometers as independent variable instead of the time. Furthermore, given that the route in both test drives is the same, the altitude over distance is also the same as shown in the black dotted line in the top plot of figure 6.9. Having been the route driven in similar conditions, the results using the two different control strategies for the TM can be compared.

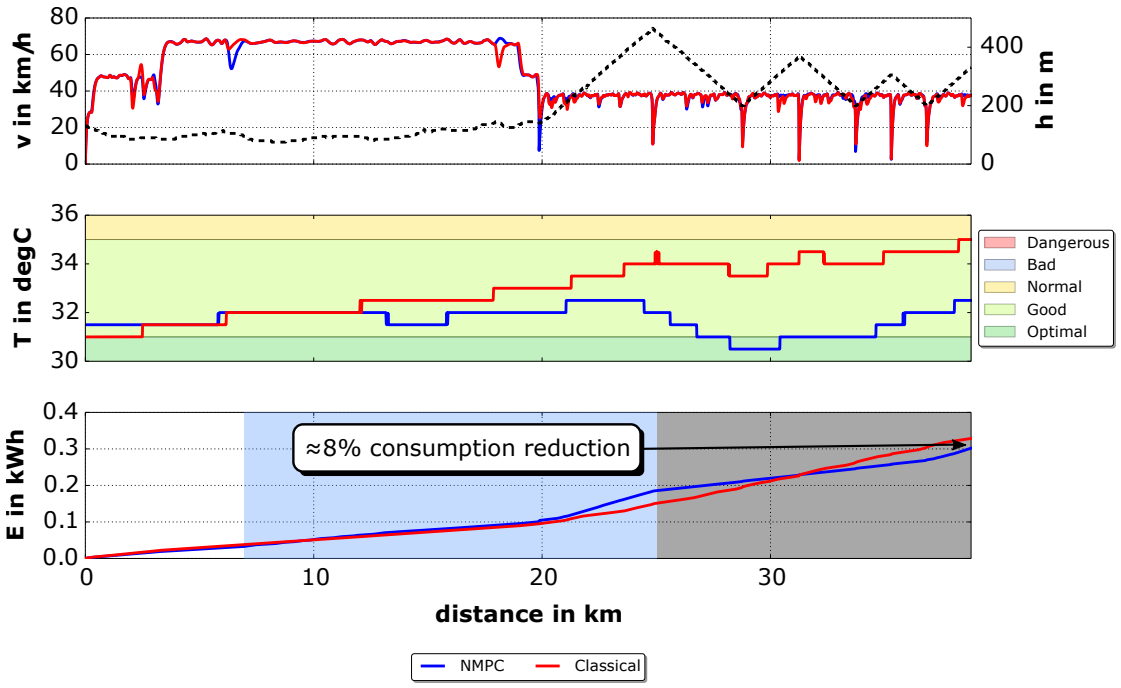


FIGURE 6.9: NMPC vs Classical TM goals in the mountain test drive under hot conditions.

Concerning the BAT temperature, the middle plot of figure 6.9 shows that the trajectory obtained with the NMPC, blue line, is in general kept within healthier ranges compared to the classical approach, red line. Indeed although the NMPC initial BAT temperature is in the "good" region and the classical in the "optimal", the NMPC manages to steer it to the "optimal" range, around kilometer 30, while the classical control approach even reaches the

"normal" region at the end of the trip.

Moreover, NMPC needs 8.14% less electrical energy for the actuators than the classical approach, as evidenced in the bottom plot of figure 6.9. Although along the blue area in the mentioned plot, the NMPC strategy needs more electrical energy than the classical control approach, in the last 25 km, black area in the figure, the predictive control achieves to compensate the previous consumption to reach the final improvement.

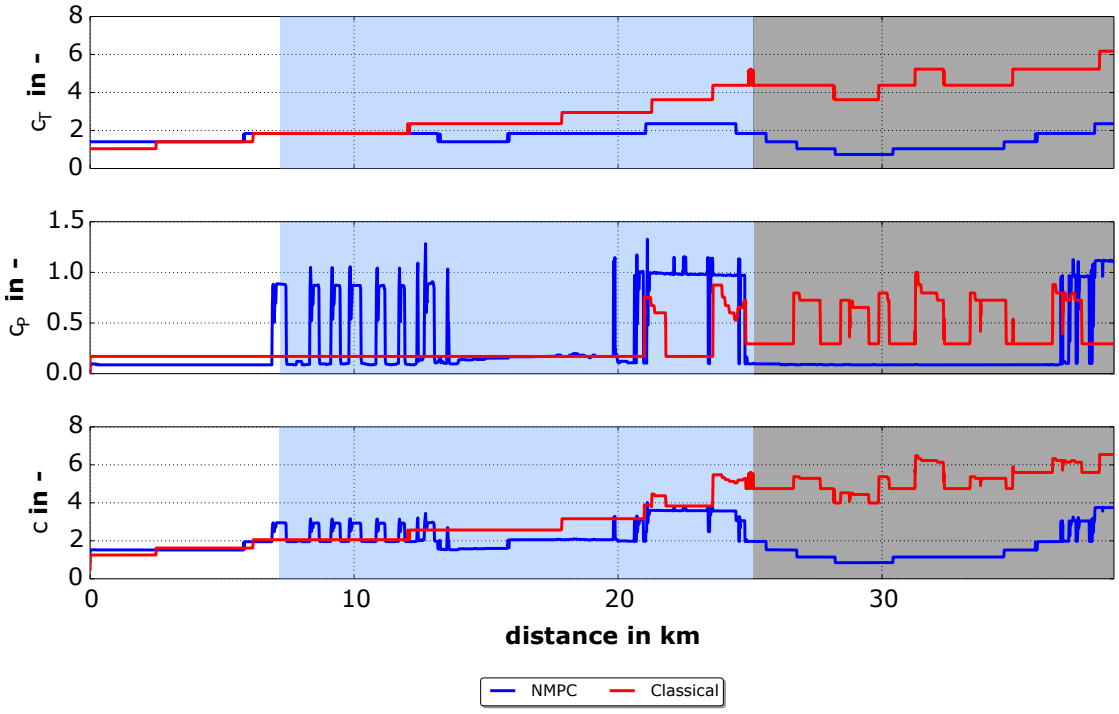


FIGURE 6.10: NMPC vs Classical TM costs in the mountain test drive under hot conditions.

Figure 6.10 represents the cost terms associated to the temperature, top plot, to the power consumption, middle plot and the global costs, bottom plot. Taking a look at the power cost term it can be seen how NMPC invests more energy resources along the blue area, thus rising the overall electrical consumption. The effect of this investment is manifest in the NMPC lower temperature costs in the top plot from the blue area till the end of the trip. In the final kilometers, black area, both costs are lower in the NMPC than in the classical control strategy.

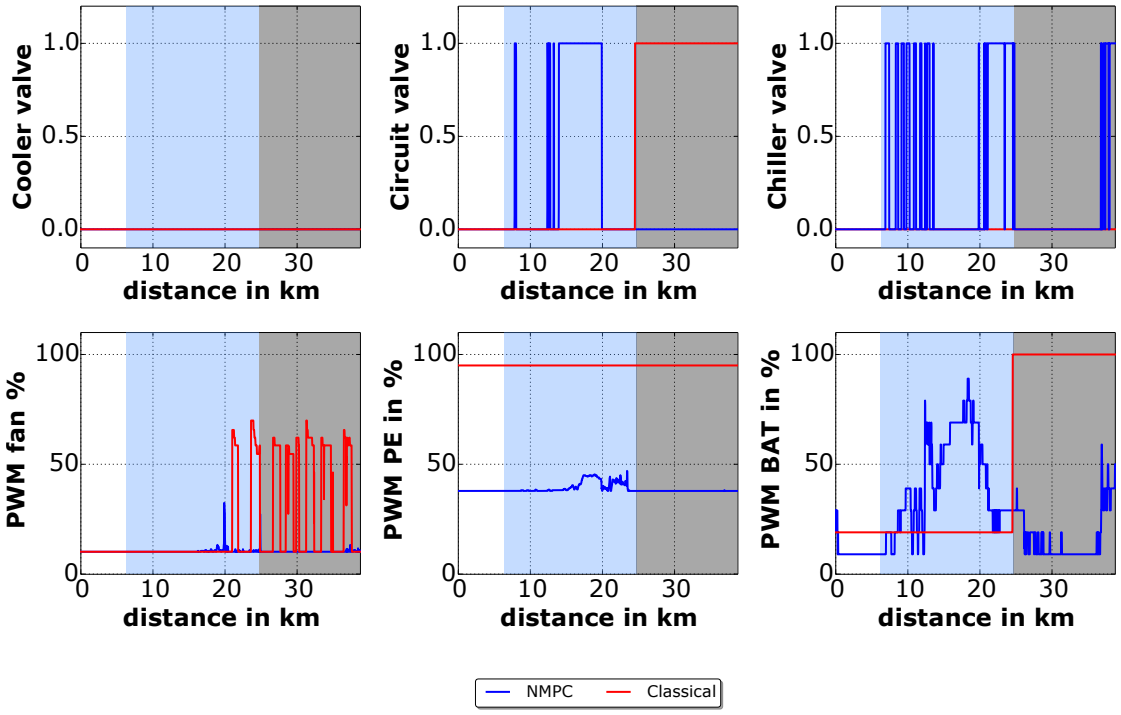


FIGURE 6.11: NMPC vs Classical TM controls in the mountain test drive under hot conditions.

However, the best way to grasp the differences of both strategies is to analyze directly the control signals used. Figure 6.11 shows the six control signals used in each test drive. Notice that due to the large temperature difference between the components and the ambient air in this test drive, figure 6.8, the cooler has a considerable potential to evacuate heat. Aware of this fact, both strategies rely on the active position of the cooler during the whole cycle, as it can be seen in the top left plot in figure 6.11.

With this cooler configuration, the classical control approach, red lines, shows two clearly differentiated working points: inside the blue area the two circuit mode is enabled, top middle plot, and the chiller is inactive, top right plot. The PWM signals for the BAT and fan, bottom left and right plots, are kept to the minimum in contrast to the PE that is at the maximum, bottom middle plot. Hence it can be said that the global strategy in the blue are is to reduce energy consumption discarding BAT cooling, since it is still in the "good" range. On the other hand, the consumption of the PE pump cannot be avoided since it is necessary for improving the component cooling through the cooler. In the black area, the previous cooling strategy based on laissez faire has lead to 34°C in the BAT and thus now an effective cooling is necessary. Therefore the one circuit mode is activated, top middle plot, the PWM signal for the BAT pump is at its maximum level, bottom right plot, and the fan, as a high consumer, is moderately used, bottom left plot.

Very different is the NMPC strategy that uses the chiller, top right plot, to cool down the BAT and also the cooler between kilometers 10 and 20, activating the one circuit mode in the middle top plot. The PWM signals of the pumps are considerably lower than the ones used in the classical control and the fan is not used at all. All in all, it can be said that the NMPC with its accurate model and the several goals and constraints is more sensitive to the current and future states of the plant and thus achieves a better TM managing the cooling resources more effectively.

6.3.2 Mountain test drive under mild conditions

The mild scenario starts with components temperatures around 22°C an the ambient air at 16°C, as it can be seen in figure 6.12. Therefore, the potential in the cooler is now lower than in the previous hot scenario. Moreover, at the beginning of the cycle the BAT is under the "optimal" temperature range and both strategies will need to heat it up. Again the temperatures plot, figure 6.12, evidences that from the thermal point of view, the NMPC and the classical test drives draw from comparable initial conditions.

Furthermore, the top plot of figure 6.13, where the red and blue lines match considerably, proofs that the mechanical conditions and indirectly the heat generated in the components is the same in both test drives.

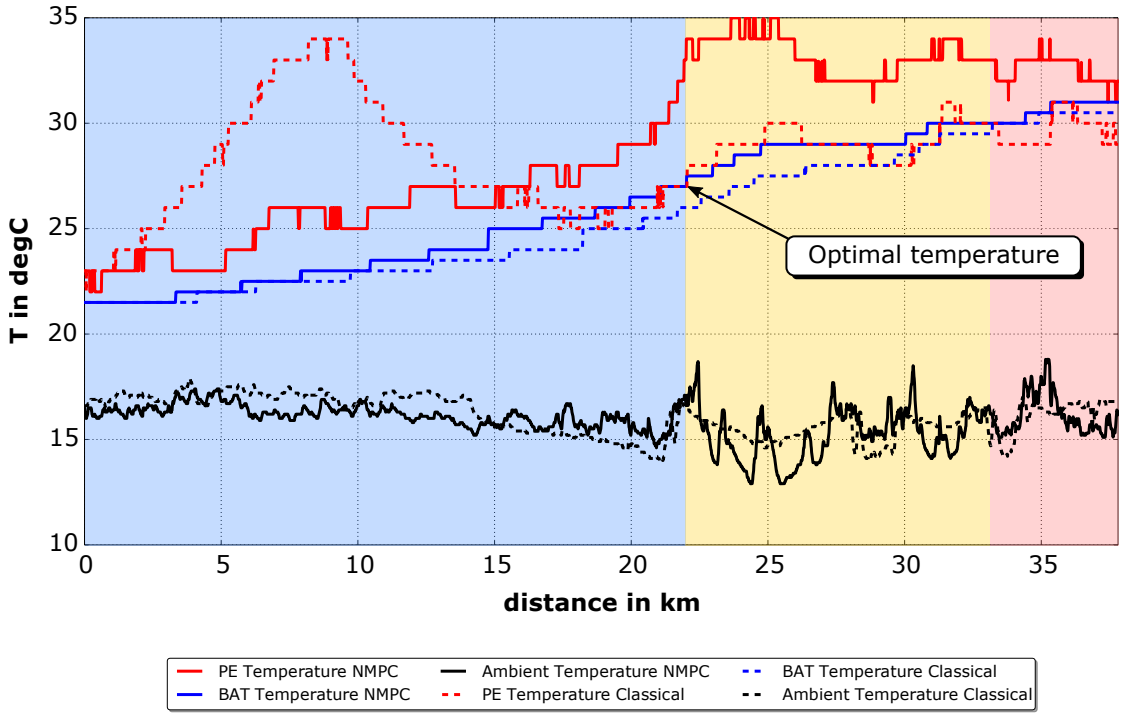


FIGURE 6.12: NMPC vs Classical TM temperatures in the mountain test drive under mild conditions.

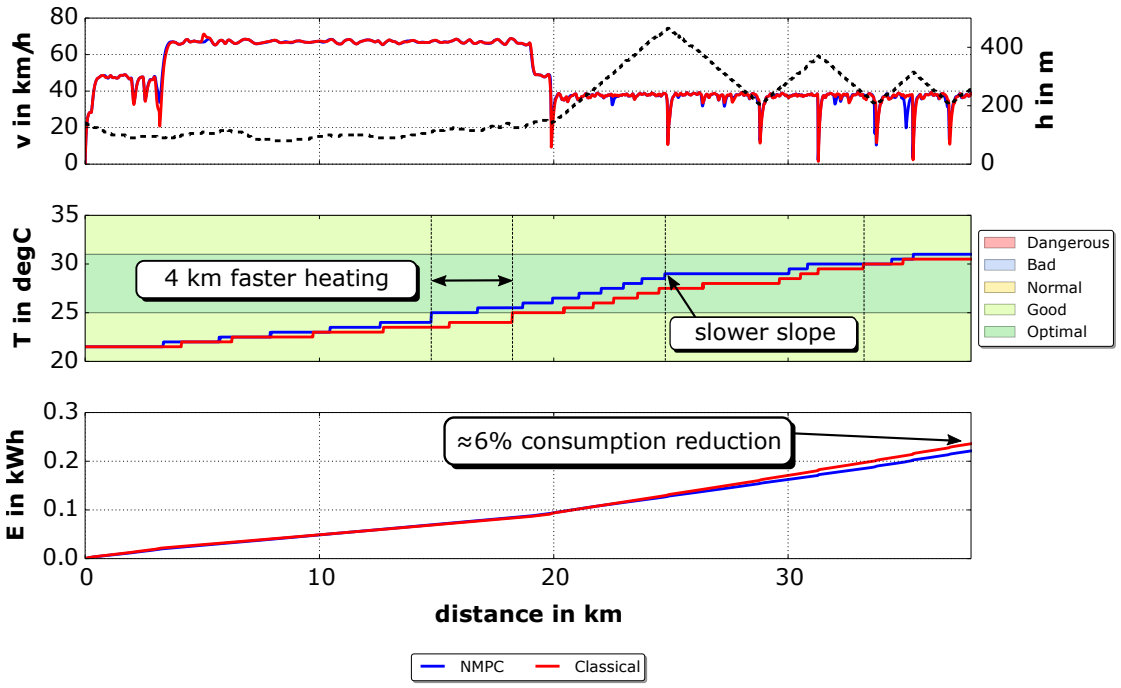


FIGURE 6.13: NMPC vs Classical TM goals in the mountain test drive under mild conditions.

Having validated that the test drives are comparable, the TM goals in figure 6.13 can be contrasted. The middle plot of the figure shows that NMPC is able to heat the BAT faster than the classical control, reaching the "optimal" range 4 km earlier. Once in the "optimal" temperatures range both strategies succeeded in keeping the BAT inside it till the end of the cycle. In other words, before entering the "optimal" range, the NMPC achieves a higher "slope" in the BAT temperature trajectory compared to the classical strategy and in addition, once inside the "optimal" range, it manages to reduce the slope even below the one in the classical strategy in order to keep the BAT temperature inside the range. Again the better temperature behavior is achieved with the consumption of less electrical energy, the NMPC needs 6.25% less energy than the classical control approach.

To understand how NMPC achieves improvements in the two goals at the same time, it is worth to take a look at the cost terms in figure 6.14 where three different parts can be distinguished:

- **1) Battery heating phase**, blue area in figure 6.14, in which the main goal is to bring the BAT temperature to the optimum as it is shown at the end of the blue area in figure 6.12. This way the costs associated to the temperature, top plot figure 6.14, are faster decreased in NMPC, blue line, than in the classical control approach, red line. The prize to pay is a slightly higher electrical consumption as represented in the middle plot.

- **2) Energy saving phase**, yellow area in figure 6.14, where being the BAT temperature in the optimum, the priority is to minimize the actuators electrical consumption as it can be seen clearly in the yellow area of the middle plot.

- **3) Battery cooling phase**, red area in figure 6.14, in which the temperature costs, this time associated to higher temperatures than the optimal, are again high enough for NMPC to invest resources.

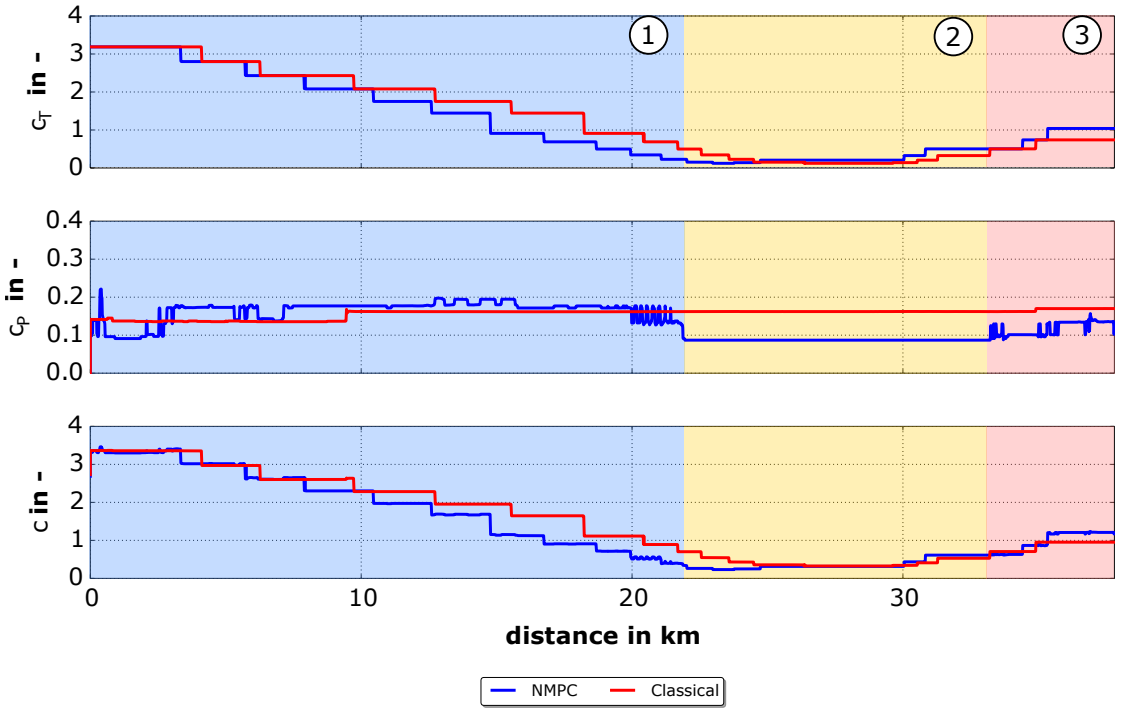


FIGURE 6.14: NMPC vs Classical TM costs in the mountain test drive under mild conditions.

Inside the different described phases, the control inputs from the NMPC strategy show a tendency as it can be seen in figure 6.15. Inside the BAT heating phase, the cooler is mostly bypassed, top left plot in the figure, since the ambient air has a lower temperature than the coolant and hence is not useful for heating the BAT. Given that the PE is a component with a lower thermal mass, the generated heat in this component is used to heat the BAT closing the system in the one circuit mode, top middle plot. The pumps, bottom middle and right plots, are used moderately and the fan and chiller are not used. In the energy saving phase, yellow area, on the contrary the valves take the position at which they do not consume energy and pumps and fan are kept at minimum PWM. Finally, in the red area, given that the BAT is getting warmer, the one circuit mode is again enabled, top middle plot, in order to transfer the heat to the fresh air through the cooler.
On the contrary, the classical control strategy does not coupled both circuit at all, red line in the top middle plot, and the only cooling effort is put on the PE as soon as it reaches 35°C

as evidences the maximum PWM signal in the bottom middle plot.

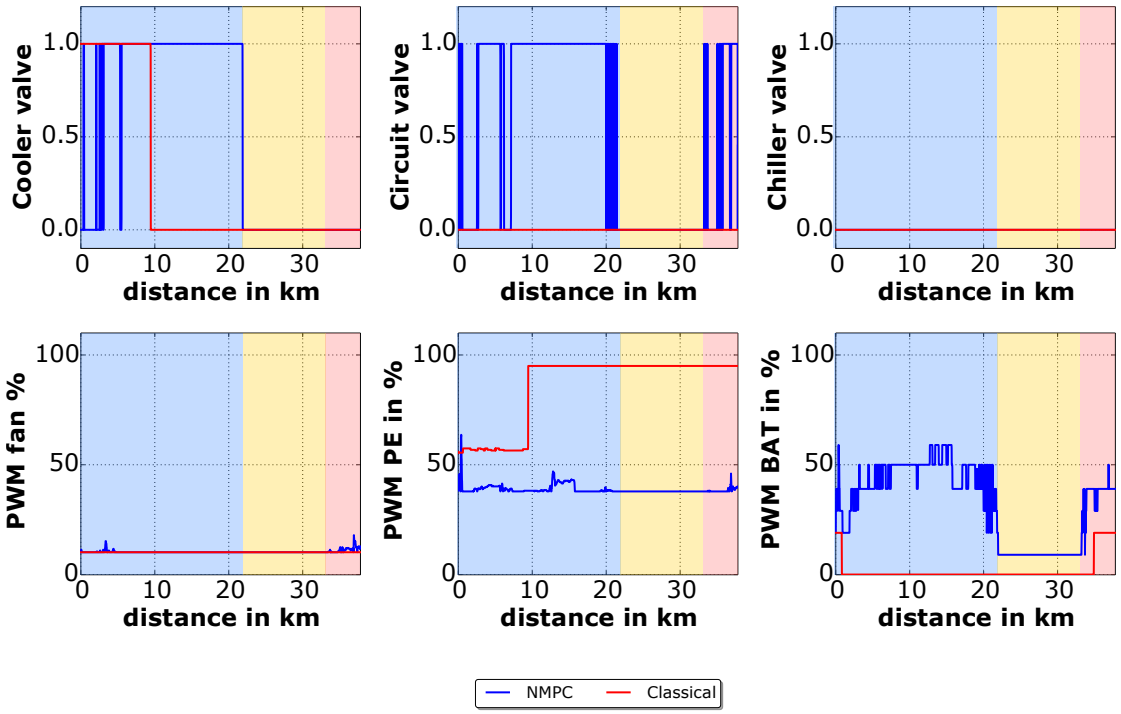


FIGURE 6.15: NMPC vs Classical TM controls in the mountain test drive under mild conditions.

6.3.3 Highway test drive under cold conditions

Analogously to the mentioned test drives, the highway driving cycle draws from comparable temperatures as shown in figure 6.16. Notice that, despite the similar initial conditions, the ambient temperature in the last 5 kilometers of the cycle presents a remarkable difference of 3°C between the NMPC and the classical control approach. Furthermore, in the last 3 kilometers, the presence of another vehicle in the road leads to the speeds discrepancy in the top plot of figure 6.17. Although these differences are not desirable for comparison, they are not accumulative: the cycle with the vehicle disturbance, the NMPC, generates slightly less heat, due to the lower speed along 3 km, but on the other hand it has less potential in the cooler given that the ambient temperature is higher than in the classical approach in the last 5 kilometers of the trip. Therefore, it can be assumed that the conditions are fair enough for comparison.

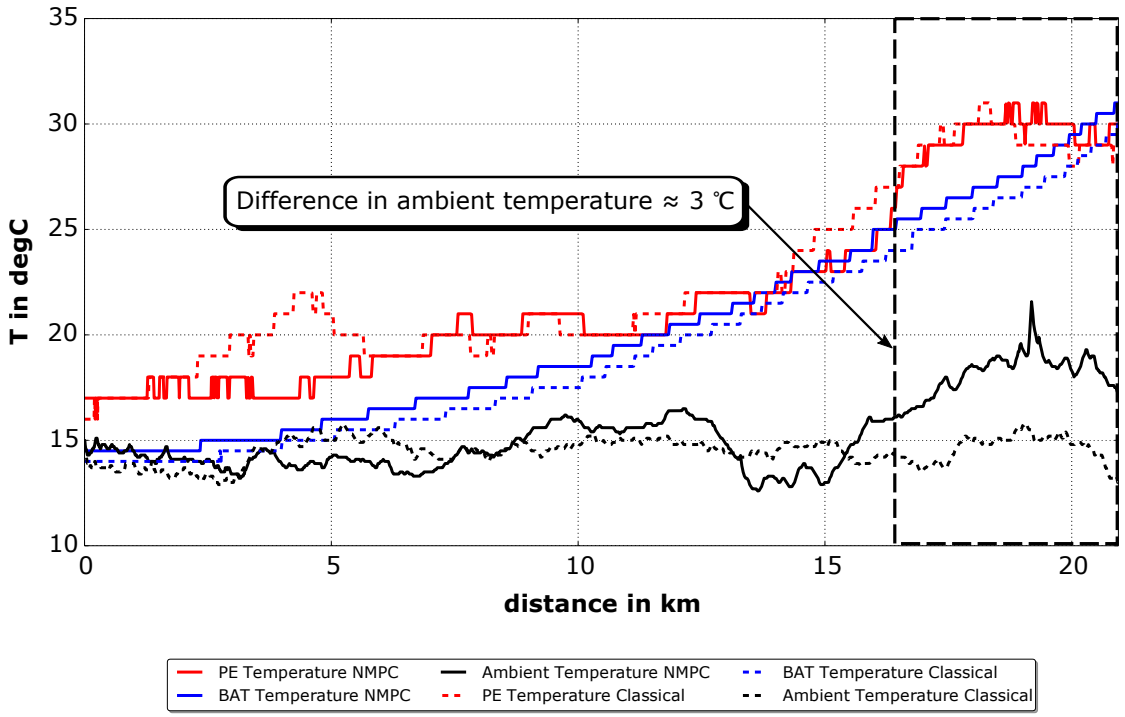


FIGURE 6.16: NMPC vs Classical TM temperatures in the highway test drive under cold conditions.

In this scenario, coolant, components and ambient air are cold, around 14°C. Therefore both TM strategies aim at heating up the BAT as shown in the middle plot of figure 6.17, being the NMPC trajectory slightly faster in this task. Nevertheless, in contrast to the previous studied cases, the NMPC fails in minimizing the two goals at the same time. In the current cycle, the better temperature performance is paid with a 3.5% higher electrical consumption in the NMPC case.

This can be seen in more detail in figure 6.18, where the top plot evidences that NMPC achieves lower costs associated to the BAT temperature and the middle plot shows that it is at the prize of higher consumption costs. At the end of the cycle, when the BAT is up to leave the "optimal" range, the NMPC even uses the chiller briefly to keep it in the range. As the bottom plot of figure 6.18 suggests, the total costs are still lower in the NMPC case, since the temperature has a more important weight in the sum.

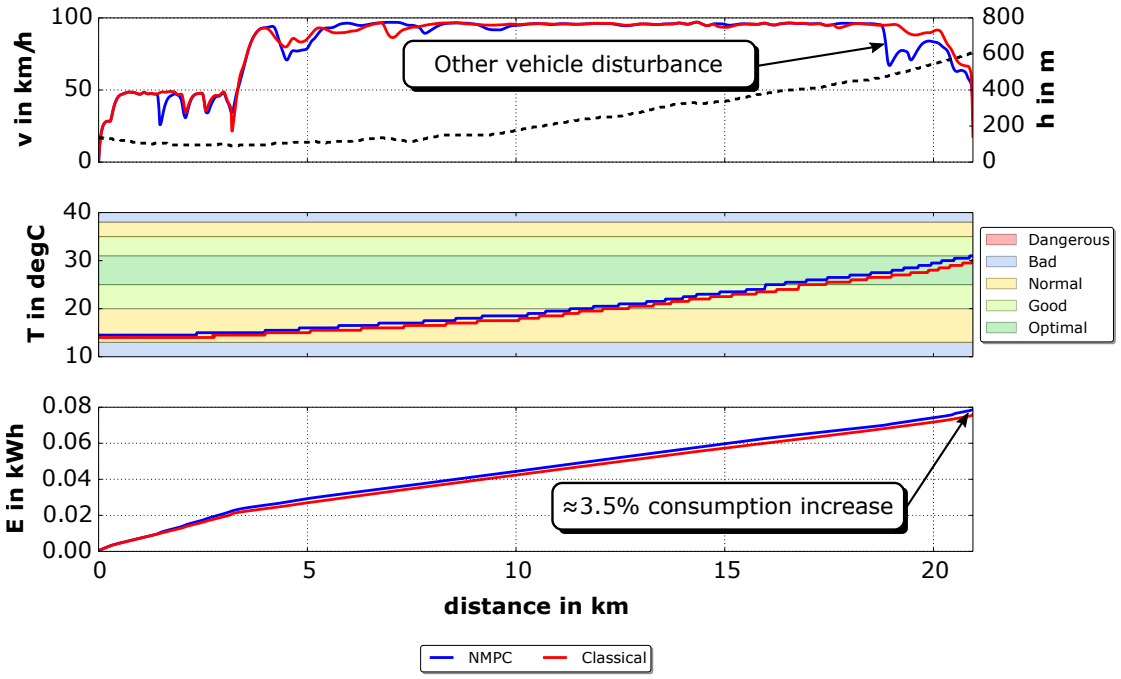


FIGURE 6.17: NMPC vs Classical TM goals in the highway test drive under cold conditions.

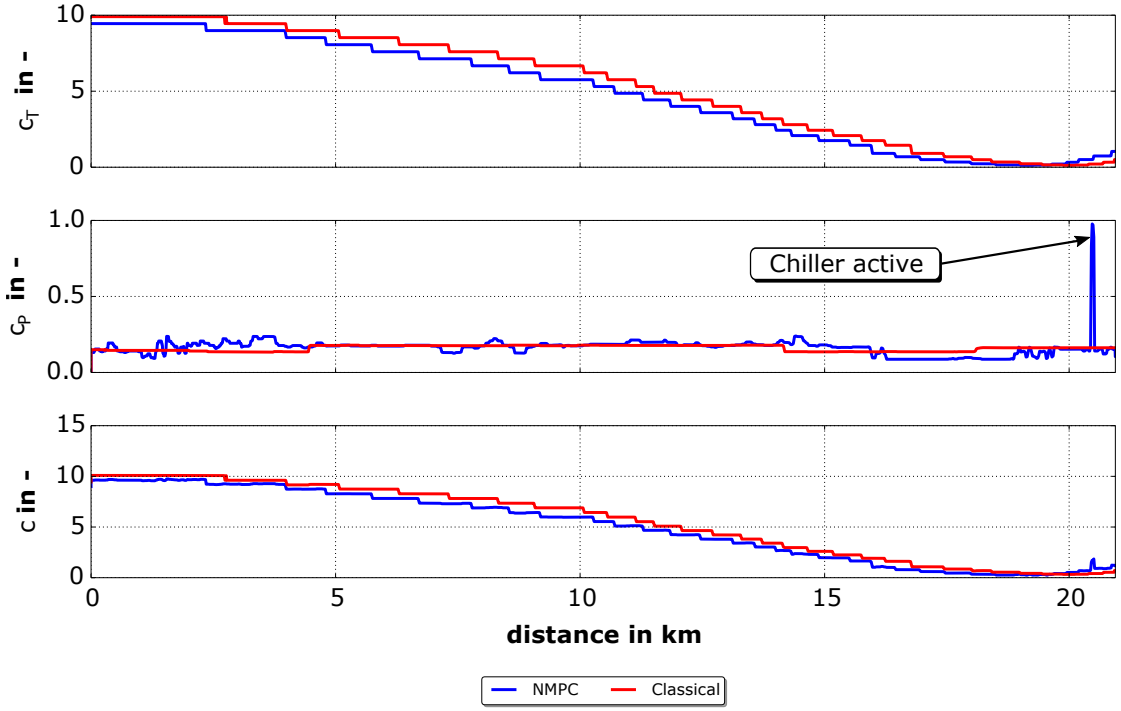


FIGURE 6.18: NMPC vs Classical TM costs in the highway test drive under cold conditions.

The controls in figure 6.19 show that the NMPC as well as the classical control approach bypass most of the trip the cooler because it is colder than the coolant and both are below the optimal temperature, top left plot in the figure. The top middle plot shows that NMPC couples the two components in the one circuit mode to use the generated heat in the PE to

heat up the BAT, while the classical control approach does it approximately the half of the cycle, being they decoupled at the beginning and the end. Both strategies refuse the use of the fan and the chiller, excepting the final attempt of NMPC to keep the BAT in the "optimal" range. Finally, notice that the PWM signal of the BAT and PE pumps is higher and lower, respectively, in the NMPC control approach.

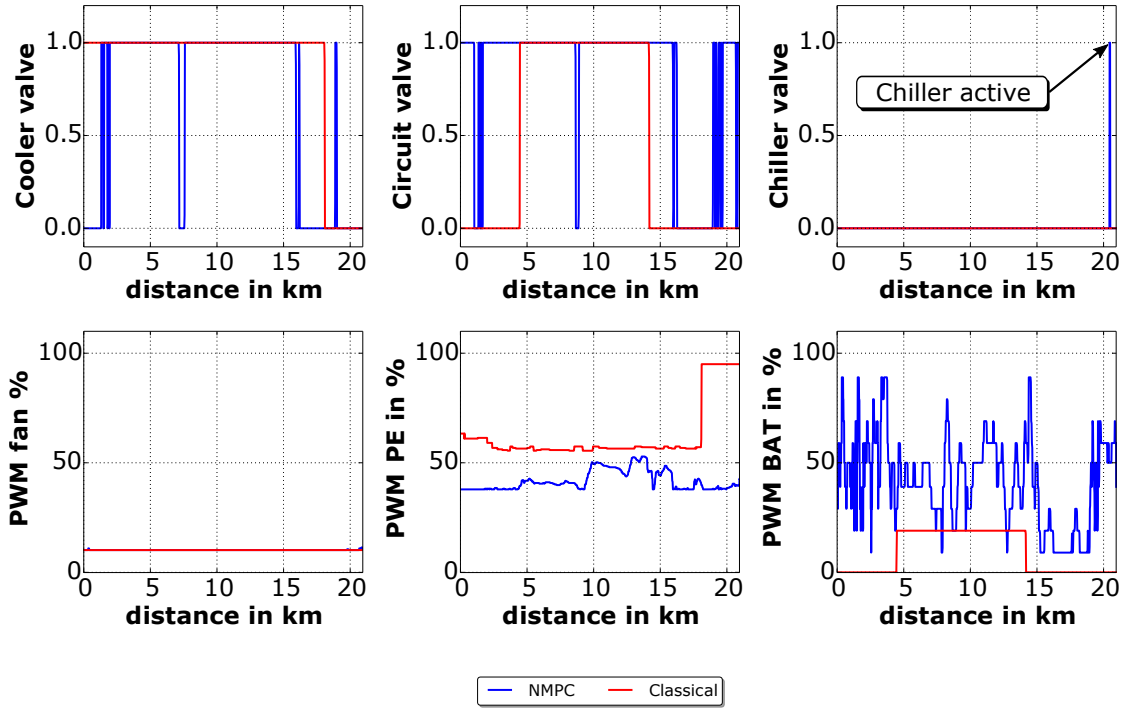


FIGURE 6.19: NMPC vs Classical TM controls in the highway test drive under cold conditions.

It must be added that the fact that this cycle is driven at constant speed, places the standard strategy in an advantageous situation, since finite-state machines are usually defined with several static points at which control experience is available. Therefore, the less transient and the more common the driving conditions are, the more accurate is this method.

Furthermore, the fact that the system has no heat sources to actively heat up the components makes that a cold scenario such as the highway test drive offers less possibilities to the cooling circuit that can only couple the components and wait till the generated heat conducts them to the "optimal" range. This is a further disadvantage for NMPC which shows its major potentials in complex systems where there are many possibilities for the controls. Nevertheless, all in all it can be concluded that even in an scenario where the classical control strategy can show its major performance, the NMPC still achieves a better TM from the total costs point of view.

6.4 Vehicle results discussion

Table 6.2 shows the main differences of the TM resulting from the two control strategies.

TABLE 6.2: TM Vehicle results.

Test drive	$\Delta\mu_{T_{BAT}}$ in °C	ΔE_{LT2} in %	ΔE_{LT2} in kWh	$\Delta\Phi$ in %
Mountain hot	-1.57	-8.14	-0.0267	-49.69
Mountain mild	0.73	-6.25	-0.0148	-9.86
Highway cold	0.78	3.49	0.0027	-8.14

The first two columns are the difference in the average BAT temperature (5.1a) and the electrical consumption of the circuit at the end of the test drive in % relative terms (5.1b) while in the third column the consumption difference is expressed in kWh and the last column stands for the relative improvement of the total costs Φ.

As it can be seen in the last column in Table 6.2, the predictive method obtains significantly lower costs in the three measured cases, having being the most remarkable costs improvement observed in the mountain test drive under hot conditions, around 50%. The mountain mild and highway cold test drives in spite of showing less potential, they still lead to costs improvements around 10% and 8%.

For a better understanding it is worth to unpack this information in the two goals separately. Taking a look at the electrical consumption, in relative and absolute terms, second and third columns in Table 6.2, it can be said that NMPC succeeded in the hot and mild scenarios, however it failed in the cold scenario. On the contrary, focusing on the temperatures obtained with both methods, first column in the table, it must be highlighted that in the three cases the predictive method overcame the classical one with temperature improvements around 1°C. In order to analyze the temperature improvement more deeply, the normal distribution of the temperatures obtained with NMPC and the classical control strategies in the hot, mild and cold scenarios are represented in figure 6.20.

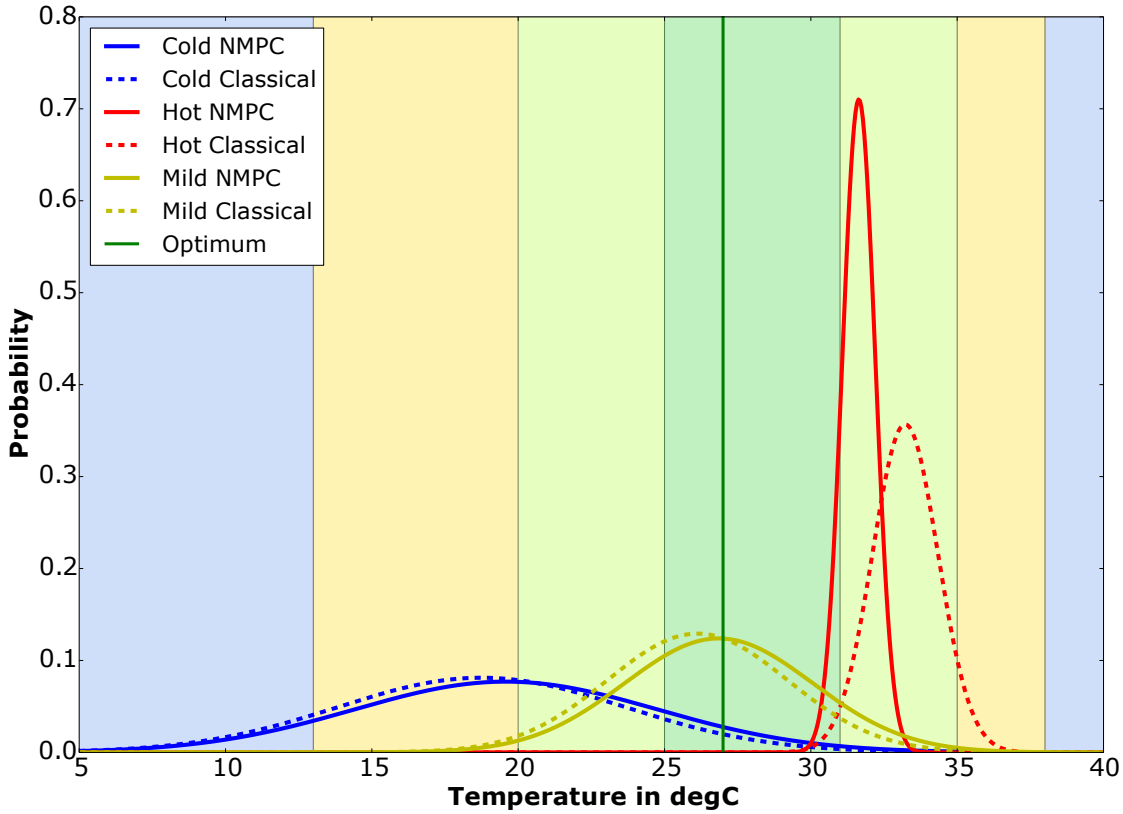


FIGURE 6.20: Temperatures normal distribution with NMPC and Classical TM controls in the three studied scenarios.

Concentrating on the blue curves in figure 6.20, it can be said that for the cold climate/stationary speed profile combination, the BAT temperatures obtained with NMPC , blue solid line, are slightly shifted closer to the optimum compared to the classical strategy, dashed blue line. Nevertheless, the shape of the Gaussian bell is very similar in both cases, evidencing that the temperatures variance is very similar.

In the mountain mild case, yellow lines, the shifting to the optimum produced by NMPC compared to the classical control strategy is more prominent than in the cold scenario while the difference in the shape of the curves, dashed versus solid, is still very similar to the cold scenario.

However, it is in the hot mountain scenario, red lines, that NMPC shows clearly a temperature regulation improvement. In addition to a clear displacement of the average temperature to the optimum, it shows a curve shape significantly narrower than the one obtained with the classical control: while NMPC keeps the temperature within the "good" and "optimal" ranges, green areas in figure 6.20, the classical control strategy leads to temperatures which stretch from the "normal" range, yellow area in figure 6.20, to the "optimal".

Thus, similarly to the simulation results it can be concluded that the potential of the designed NMPC is higher at hot temperatures, where the situation becomes more and more complex:

several heat sources and heat sinks are at stake and it is not trivial to manage the many resources to minimize the two goals in the cost function. By means of a more intensive use of the valves and pumps and the preference of using the chiller instead of the fan, due to its higher efficiency, the NMPC permits a considerable TM improvement in the hot scenario.
At the other end, in a cold scenario the studied circuit offers few possibilities to warm up the BAT, since the heat can only be obtained from the air, in case it is warmer, or from the PE circuit given that this component has a lower thermal mass. In addition to the system "simplicity" in this scenario, the fact that this test drive was performed at constant speed, supposes a further challenge for NMPC to overcome a classical strategy that is specially suitable for static and common operation points. Despite these obstacles, it must be highlighted that NMPC still achieves an improvement in the total costs.
Concerning this scenario it must be also added that although the overall cost is satisfactory, the electrical consumption has not been reduced by NMPC, red numbers in Table 6.2. If additionally both goals should be improved at the same time, it would be quite straightforward to achieve adjusting the cost function parameters c_T and c_P. This is a further advantage in comparison with a PID tuning process, where the effect of the P, I and D gains on the several goals are not so intuitively and directly attributable to them.
All in all, we conclude that in the three measured real scenarios NMPC has shown to be capable of grasping the untapped potential of the classical control strategy and furthermore we have observed that the more degrees of freedom the control of the plant offers, the higher is the untapped potential to be exploited by NMPC.

Chapter 7

Conclusions and further works

7.1 Conclusions

In this thesis a real-time NMPC for the TM of the BAT and PE cooling circuit in a PHEV has been proposed. The main conclusions drawn in this project can be grouped according to the following topics.

NMPC for TM

The proposed NMPC has been validated by means of six simulations and three real test drives characterizing several driving scenarios and heterogeneous climate conditions. In order to numerically evaluate the effect of NMPC on the TM, a finite-state machine-based control strategy has been developed for the simulation comparison while on the road the same driving cycle has been performed with the predictive method and with the original control implemented in the vehicle.

In five of the six simulation cases and in all vehicle test drives, NMPC has shown an improvement in the simultaneous achievement of the two defined TM goals compared to the classical control approach. Furthermore, both test environments agree that the more challenging the thermal situation is, the better results derive from applying NMPC. Hence under mild and hot temperatures, the NMPC approach in simulation has shown average BAT temperatures around 1 and 3°C closer to the optimum than the classical control achieving at the same time electrical consumption reductions between 9 and 14%. The same tendency has been finally observed in the performed mild and hot test drives where the consumption improvement between 6 and 8% goes hand in hand with a healthier temperature profile. The main reason for these improvements is that in contrast to the classical control strategy which is based on several defined working points, NMPC goes through many more combinations of the controls

basing the decisions on its reliable model and optimization algorithm. Thus, the valves are switch more often, the pumps work at numerous PWM levels and a more flexible use of the high consumers, chiller and fan, according to their efficiency is possible.

Less potential has been observed in a cold climate scenario, where in the real vehicle measurements despite the overall TM costs improvement, the NMPC had to pay 3.5% more electrical energy to achieve a faster warming up of the BAT. In the simulation two driving scenarios under cold climate conditions were validated. One of them succeeded in both goals achievement and the other one failed in improving the temperature profile even leading to an increase of the global costs. The argument for these results is that in a cold scenario, the system offers few possibilities to the control for achieving the main goal of heating up the BAT. For this purpose, the cooling circuit does not incorporate any external heating source and can only offer the air flowing through the cooler, in case it is warmer, or the use of the in the PE generated heat, which is faster available due to the smaller thermal mass of this component compared to the BAT. Therefore, in this scenario a classical control method does not leave a high untapped potential to be exploited by NMPC. Nevertheless, it is important to highlight that under these conditions in the real test drive, NMPC still achieves to reduce the overall costs.
It must be also highlighted that in all studied cases NMPC besides minimizing the cost function, it has fulfilled the temperature constraints, thus keeping the PE successfully under 60°C.

On the other hand, another appealing feature in NMPC has been shown, which is its calibration process. Usually, classical control methods rely on PID controllers which need to be adjusted. The process is based on varying the P, I and D gains and evaluating its influence on the several goals where it is often complex to attribute to the gains variations the effect on the multiple goals. As seen in this thesis, NMPC requires also a calibration of the costs parameters associated to the different goals. Nevertheless, in contrast to a PID controller tuning method, the calibration in NMPC is more natural and straightforward since, in a few words, it consists of weighting up the goals.

Modeling for NMPC

The development of a model with an accuracy/simplicity trade-off is crucial for NMPC. The modeling approach taken here has shown to capture the system dynamics, with a temperature error under 1°C for the BAT and 2°C for the PE, and at the same time to be fast enough for real-time performance. For this success the hybrid approach of combining physical equations with experimental data in an object oriented language describing the multidisciplinary processes that govern the controlled plant has been crucial. Another key aspect for modeling such a complex thermal system in a relative short time has been the use of the TILMedia

library for Modelica from TLK-Thermo GmbH which facilitates efficient thermal properties calculations.

Real-time NMPC

As said at the beginning of this thesis, currently NMPC methods are gaining attention for fast applications. In an automobile there are numerous systems where NMPC could be used to exploit their untapped potential. However, the bottleneck for the widespread adoption of the method in the sector is the associated computational burden. In this context, with the present thesis it has been proven that TM, due to its slower inertia compared to most control systems in the vehicle, is a suitable field for real-time NMPC with the current resources.
The success in achieving real-time performance is mainly due to the efficient and robust algorithms behind the used optimization tool MUSCOD-II which as well in the vehicle as in simulations, has shown a stable behavior. On the other hand, it must be highlighted that besides using a powerful numerical method, the suitable modeling, problem formulation (goal function design, constraints definition) and optimizer calibration have been crucial steps for achieving real-time performance.
Additionally, this thesis has contributed in the NMPC implementation in the automotive sector proposing a rapid-prototyping-based methodology for moving from simulation to the real vehicle.

NMPC future: a personal prediction

Despite the great potential of optimization-based methods, similarly to the transition from conventional cars to E-Mobility, the change in the control methodology is not going to happen overnight. Usually several technical departments are behind the design, implementation and testing of control systems, collaborating closely and distributing the tasks to fulfill the many requirements. As a consequence, particular and specific know how is built in the different departments. For the implementation of a centralized control system for multi-domain processes such as the NMPC method shown in this thesis, the role of a generalist engineer capable of condensing the knowledge of the several specialists in a valid model and optimal control formulation is crucial. Furthermore, an increase of simulation and numerical methods experts are needed since they play an important role similarly as technicians do in the test bench.
Therefore, until an important investment in human capital and tools is made, NMPC will progress moderately in form of small R&D projects such as the one presented here.

However, an alternative for a first shot in optimization tools widespread could be to promote their off-line use on the desk with the aim of identifying new working points and rules that

improve the current control logic. Thus, through this support task prior to the implementation in the vehicle, the optimization tools can grow in familiarity and trust in the sector, paying the prize of not exploiting its entire potential which is only shown in on-line performance.

7.2 Further works

Despite the fulfillment of the aims and objective defined in the first chapter, during the realization of this project several topics were found which would be interesting to study in future investigations.

Future information

One of the positive features of NMPC methods is that they can use available future information to improve their predictions and thus the results. In this thesis, it was assumed that the mechanical variables speed, torque and rotational speed do not vary within the prediction horizon. Nevertheless, future information concerning these variables would lead to better predictions of the generated heat in the system and consequently to better TM results. The best way to improve the control, would be to include a simple mechanical model that given the vehicle speed and the road slope could calculate the mechanical variables. The prediction of the vehicle speed could then be obtained from methods like [21] and the road slope prediction from the Global Positioning System (GPS).

Mixed-integer formulation

As mentioned before, one of the main simplifications taken here was the rounding of the solenoid valves control signals in order to deal with a continuous OCP. Nevertheless, despite the success obtained, to assure optimality it would be necessary to study the MIOCP formulation and the methods associated to it. Given that most mixed-integer algorithms lead to higher computational burden, the authors recommend to take a look at the fast methods shown in [70].

Variable calibration

In the simulation results shown, different cost function parameters were obtained for a suitable performance in the six studied driving situations. Motivated by this, prior to the vehicle implementation, it could be useful to measure a range of possible scenarios for calibrating the cost function parameters and store them in look-up tables in advance. Hence, the vehicle could use variable cost function parameters to obtain a further improvement in the results.

Climate chamber

Despite the experiments were successfully designed with the aim of being comparable, for a more accurate comparison, especially in temperature terms, it would be advisable to test the method in a climate chamber where both control strategies would face exactly the same situation.

Controlled plant states feedback

In this project, the studied cooling circuit was equipped with fifteen temperature sensors to output the state of the controlled plant required by the optimizer. Nevertheless, an increasing number of sensors implies higher costs and hence for the final vehicle implementation it would be desirable to avoid them. In this direction, it would be interesting to study methods like the extended Kalman filter or moving horizon estimation which can estimate the states of nonlinear plants without the need of including extra sensors. A comparison of the mentioned methods can be found in [79].

Bibliography

[1] Various. *Systems perspectives on electromobility*. Ed. by Björn Sandén and Pontus Wallgren. Chalmers University of Technology, 2014. ISBN: 9789198097399.

[2] Barney Cohen. "Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability". In: *Technology in Society* 28.1-2 (2006), pp. 63–80. ISSN: 0160791X. DOI: 10.1016/j.techsoc.2005.10.005.

[3] Various. *Electric vehicles in Europe : gearing up for a new phase?* Tech. rep. Amsterdam: Amsterdam Roundtables Foundation and McKinsey & Company, 2014.

[4] Auvinen Heidi et al. "Process to support strategic decision - making : Transition to electromobility". In: *EVS27 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*. 2013, pp. 1–8. ISBN: 9781479938322. DOI: 10.1109/EVS.2013.6914712.

[5] Masataka Wakihara. "Recent developments in lithium ion batteries". In: *Materials Science and Engineering R33* February (2001), pp. 109–134.

[6] Eckhard Karden et al. "Energy storage devices for future hybrid electric vehicles". In: *Journal of Power Sources* 168 (2007), pp. 2–11. DOI: 10.1016/j.jpowsour.2006.10.090.

[7] Vinodkumar Etacheri et al. "Challenges in the development of advanced Li-ion batteries : a review". In: *The Royal Society of Chemistry* Energy and Environmental Science (2011). DOI: 10.1039/c1ee01598b.

[8] *Tesla Model 3*. URL: https://www.teslamotors.com/model3 (visited on 03/05/2016).

[9] Victor Agubra and Jeffrey Fergus. "Lithium Ion Battery Anode Aging Mechanisms". In: *Materials* 6 (2013), pp. 1310–1325. DOI: 10.3390/ma6041310.

[10] Sarah J. Gerssen-Gondelach and André P.C. Faaij. "Performance of batteries for electric vehicles on short and longer term". In: *Journal of Power Sources* 212.August (2012), pp. 111–129. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2012.03.085. URL: http://linkinghub.elsevier.com/retrieve/pii/S0378775312007069.

[11] J Shim et al. “Electrochemical analysis for cycle performance and capacity fading of a lithium-ion battery cycled at elevated temperature”. In: *Journal of Power Sources* 112 (2002), pp. 222–230.

[12] Thomas H. Dubaniewicz and Joseph P. Ducarme. “Are lithium ion cells intrinsically safe?” In: *IEEE Transactions on Industry Applications* 49.6 (2013), pp. 2451–2460. ISSN: 00939994. DOI: 10.1109/TIA.2013.2263274.

[13] S S Zhang, K Xu, and T R Jow. “The low temperature performance of Li-ion batteries”. In: *Journal of Power Sources* 115 (2003), pp. 137–140.

[14] Lluc Canals-casals. “El envejecimiento de las baterías de un vehículo eléctrico y cómo lo percibe el conductor The electric vehicle battery ageing and how it is perceived by its drivers”. In: *Dyna* 91 (2016), pp. 188–195.

[15] Languang Lu et al. “A review on the key issues for lithium-ion battery management in electric vehicles”. In: *Journal of Power Sources* 226 (2013), pp. 272–288. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2012.10.060. URL: http://dx.doi.org/10.1016/j.jpowsour.2012.10.060.

[16] Huai Wang et al. “Transitioning to Physics-of-Failure as a Reliability Driver in Power Electronics”. In: *IEEE Journal of emerging and selected topics in power electronics* 2.1 (2014), pp. 97–114.

[17] D Antonijevic and R Heckt. “Heat pump supplemental heating system for motor vehicles”. In: *Proceedings of the Institution of Mechanical Engineers, Part C*. Vol. 218. 2004, pp. 2–6.

[18] K Y Kim, S C Kim, and M S Kim. “Experimental studies on the heating performance of the PTC heater and heat pump combined system in fuel cells and electric vehicles”. In: *International Journal of Automotive Technology* 13.6 (2012), pp. 971–977. DOI: 10.1007/s12239.

[19] Ahmad A. Pesaran. “Battery Thermal Management in EVs and HEVs : Issues and Solutions”. In: *Advanced Automotive Battery Conference*. Las Vegas, Nevada, 2001. ISBN: 0005-6359.

[20] M Hoekstra et al. “Health technology assessment review: Computerized glucose regulation in the intensive care unit - how to create artificial control”. In: *Critical Care* (2009). DOI: 10.1186/cc8023.

[21] J. J. Valera et al. “Driving cycle and road grade on-board predictions for the optimal energy management in EV-PHEVs”. In: *EVS27 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium* (2013), pp. 1–10. DOI: 10.1109/EVS.2013.6914763.

[22] S.Joe Qin and Thomas a. Badgwell. “A survey of industrial model predictive control technology”. In: *Control Engineering Practice* 11.7 (2003), pp. 733–764. ISSN: 09670661. DOI: 10.1016/S0967-0661(02)00186-7.

[23] Jay H. Lee. “Model predictive control: Review of the three decades of development”. In: *International Journal of Control, Automation and Systems* 9.3 (2011), pp. 415–424. ISSN: 1598-6446. DOI: 10.1007/s12555-011-0300-6. URL: http://link.springer.com/10.1007/s12555-011-0300-6.

[24] D Q Mayne et al. “Constrained model predictive control : Stability and optimality”. In: *Automatica* 36 (2000), pp. 789–814. URL: www.elsevier.com/locate/automatica.

[25] Vibhor L Bageshwar, William L Garrard, and Rajesh Rajamani. “Model Predictive Control of Transitional Maneuvers for Adaptive Cruise Control Vehicles”. In: *IEEE Transactions on Vehicular Technology* 53.5 (2004), pp. 1573–1585.

[26] Roman Schmied et al. “Nonlinear MPC for Emission Efficient Cooperative Adaptive Cruise Control”. In: *5th IFAC Conference on Nonlinear Model Predictive Control*. 2015.

[27] Thomas Stanger and Luigi del Re. “A model predictive Cooperative Adaptive Cruise Control approach”. In: *2013 American Control Conference* (2013), pp. 1374–1379. DOI: 10.1109/ACC.2013.6580028. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6580028.

[28] Mike Huang, Hayato Nakada, and Ken Butts. “Nonlinear Model Predictive Control of a Diesel Engine Air Path : A Comparison of Constraint Handling and Computational Strategies”. In: *5th IFAC Conference on Nonlinear Model Predictive Control*. 2014. Seville, 2015.

[29] J Sowman D S Laila and A J Cruden P Fussey. “Nonlinear Model Predictive Control for Cold Start Selective Catalytic Reduction”. In: *IFAC-PapersOnLine* 48 (2015), pp. 471–476. DOI: http://dx.doi.org/10.1016/j.ifacol.2015.11.32. URL: http://www.sciencedirect.com/science/article/pii/S2405896315026075.

[30] Tl McKinley and Ag Alleyne. “Adaptive Model Predictive Control of an SCR Catalytic Converter System for Automotive Applications”. In: *IEEE Transactions on Control Systems Technology* 20.6 (2012), pp. 1533–1547. URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6065770.

[31] H Ali Borhan et al. “Nonlinear Model Predictive Control for Power-split Hybrid Electric Vehicles”. In: *49th IEEE Conference on Decision and Control*. Atlanta, 2010, pp. 4890–4895. ISBN: 9781424477463.

[32] Zhang Jiangyan et al. “Nonlinear MPC-Based Power Management Strategy for Plug-in Parallel Hybrid Electrical Vehicles”. In: *Proceedings of the 33rd Chinese Control Conference*. 2014.

[33] Martina Josevski and D Abel. “Flatness-based Model Predictive Control for the Fuel Optimization of Hybrid Electric Vehicles”. In: *5th IFAC Conference on Nonlinear Model Predictive Control*. Seville, 2015.

[34] Fengjun Yan, Junmin Wang, and Kaisheng Huang. “Torque-Split Strategy Incorporating Engine Transient Characteristics”. In: *IEEE Transactions on Vehicular Technology* 61.6 (2012), pp. 2458–2467.

[35] Chunhua Zheng, Guoqing Xu, and Yimin Zhou. “Realization of PMP-based Power Management Strategy for Hybrid Vehicles Based on MPC Scheme”. In: *4th IEEE International Conference Information Science and Technology (ICIST)*. 2014, pp. 682–685. ISBN: 9781479948086. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={\&}arnumber=6920569{\&}isnumber=6920317.

[36] Alicia Arce et al. “Real-Time Implementation of a Constrained MPC for Efficient Airflow Control in a PEM Fuel Cell”. In: *IEEE Transactions on Industrial Electronics* 57.6 (2010), pp. 1892–1905.

[37] Niels Van Duijkeren, Tamas Keviczky, and Peter Nilsson. “Real-Time NMPC for Semi-Automated Highway Driving of Long Heavy Vehicle Combinations”. In: *5th IFAC Conference on Nonlinear Model Predictive Control*. Seville, 2015.

[38] Mooryong Choi and Seibum B Choi. “Model Predictive Control for Vehicle Yaw Stability With Practical Concerns”. In: *IEEE Transactions on Vehicular Technology* 63.8 (2014), pp. 3539–3548.

[39] M. Bruckner et al. “Predictive thermal management of combustion engines”. In: *Proceedings of the IEEE International Conference on Control Applications* (2006), pp. 2778–2783. DOI: 10.1109/CACSD-CCA-ISIC.2006.4777078.

[40] Hasan Esen et al. “Cabin Heat Thermal Management in Hybrid Vehicles using Model Predictive Control”. In: *22nd Meditteranean Conference on Control and Automation (MED)*. Palermo, 2014. ISBN: 9781479959013.

[41] Emanuel Feru et al. “Model Predictive Control of a Waste Heat Recovery System for Automotive Diesel Engines”. In: *18th International Conference on System Theory*. Romania, 2014, pp. 658–663. ISBN: 9781479946013.

[42] Pawel Majecki, Gerrit M Van Der Molen, and Michael J Grimble. “Real-Time Predictive Control for SI Engines Using Linear Parameter-Varying Models”. In: *5th IFAC Conference on Nonlinear Model Predictive Control*. Seville, 2015.

[43] Bart Saerens et al. “Model Predictive Control of Automotive Powertrains - First Experimental Results”. In: *47th IEEE Conference on Decision and Control*. Cancun, Mexico, 2008, pp. 5692–5697. ISBN: 9781424431243. DOI: 10.1109/CDC.2008.4738740.

[44] K Bennion and M Thornton. “Integrated Vehicle Thermal Management for Advanced Vehicle Propulsion Technologies Preprint”. In: *SAE 2010 World Congress*. February. Michigan, 2010.

[45] X Duan and G F Naterer. “International Journal of Heat and Mass Transfer Heat transfer in phase change materials for thermal management of electric vehicle battery modules”. In: *International Journal of Heat and Mass Transfer* 53.23-24 (2010), pp. 5176–5182. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2010.07.044. URL: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2010.07.044.

[46] Christian Schröder et al. “Nonlinear Model Predictive Control for Thermal and Electrical Power Management for Parallel Hybrid Electric Vehicles”. In: *Hybrid and Electric Vehicles 11th Symposium*. 2014.

[47] Hong Chen et al. “Applying Model Predictive Control in Automotive”. In: *Proceedings of the 10th World Congress on Intelligent Control and Automation*. 2012. ISBN: 9781467313988.

[48] Christian Hoffmann et al. *MUSCOD-II user manual*. Heidelberg, 2010.

[49] Stuart Birch. *No Title*. URL: http://articles.sae.org/13535/.

[50] C. Thiel et al G.Pasaoglu, D.Fiorello, A.Martino, G.Scarcella, A.Alemanno, A. Zubaryeva. *Driving and parking patterns of European car drivers -a mobility survey* . Tech. rep. Joint Research Centre European Comission, 2012. DOI: 10.2790/7028.

[51] Jelden(Volkswagen AG) H et al. “The Plug-In Hybrid of the Volkswagen Modular Transverse Matrix”. In: *MTZ* 75.Development Alternative Drives (2014), pp. 18–25.

[52] Luigi Del Re, Peter Ortner, and Daniel Alberer. “Chances and challenges in automotive predictive control”. In: *Lecture Notes in Control and Information Sciences*. Ed. by Luigi Del Re. Vol. 402. Springer, 2010, pp. 1–22. DOI: 10.1007/978-1-84996-071-7{_}1.

[53] *Dynamics of Bicycle Model in LabVIEW and ADAMS*. 2015. URL: http://www.ni.com/white-paper/13020/en/ (visited on 04/14/2016).

[54] H G Bock and K J Plitt. “A multiple shooting algorithm for direct solution of optimal control problems”. In: *Proceedings 9th IFAC World Congress Budapest*. Vol. XLII. 1984, pp. 243–247. URL: http://www.csa.com/partners/viewrecord.php?requester=gs{\&}amp;collection=TRD{\&}amp;recid=N8411827AH.

[55] Christoph C. Richter. “Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems”. PhD Thesis. TU Braunschweig Fakultät für Maschinenbau, 2008.

[56] *TIL Media Suite Library*. URL: http://www.tlk-thermo.com/index.php/de/softwareprodukte/tilmedia-suite (visited on 02/05/2016).

[57] Manuel Tetschke. “Nichtlineare Modell-prädiktive Regelung am Beispiel eines PKW-Kühlkreislaufes”. Master Thesis. Otto-von-Guericke Universität Magdeburg, 2014.

[58] William T Vetterling et al. *Numerical Recipes in C The Art of Scientific Computing*. Second. Cambridge University Press, 2002. ISBN: 0521431085.

[59] Dejan Dovzan et al. “Multi-domain modelling -an advantage in modelling, simulation and control education”. In: *Proceedings 23rd European Conference on Modelling and Simulation*. Ed. by Javier Otamendi et al. 2008. ISBN: 9780955301889.

[60] Dynasim AB. *Dymola Users Manual Version 5.3a*. Lund, Sweden, 2004. URL: http://www.dynasim.com.

[61] Modelica Association. *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification*. Tech. rep. Modelica Association, 2010. URL: http://www.modelica.org.

[62] Alexey Vdovin. “Cooling performance simulations in GT-Suite”. Master Thesis. Chalmers University of Technology, 2010.

[63] Peter von Böckh and Thomas Wetzel. “Erzwungene Konvektion”. In: *Wärmeübertragung Grundlagen und Praxis*. Springer, 2011. Chap. 3. ISBN: 978-3-642-15958-9. DOI: 10.1007/978-3-642-15959-6. URL: http://link.springer.com/10.1007/978-3-642-15959-6.

[64] Michel Andr. *Real-world driving cycles for measuring cars pollutant emissions – Part A: The ARTEMIS European driving cycles*. Tech. rep. Cedex: Institut National de Recherche sur les Transports et leur Securite, 2004.

[65] T J Barlow et al. *A reference book of driving cycles for use in the measurement of road vehicle emissions*. Tech. rep. Department for Transport TRL Limited, 2009.

[66] Jorge Nocedal, Stephen J Wright, and Stephen M Robinson. *Numerical Optimization*. Ed. by Peter; Glynn and Stephen M.Robinson. Springer, 1999. ISBN: 0387987932.

[67] M Gräber et al. “Using Functional Mock-up Units for Nonlinear Model Predictive Control”. In: *Proceedings of the 9th International Modelica Conference*. Munich, 2012. DOI: 10.3384/ecp12076781.

[68] Various. *Functional Mock-up Interface for Model Exchange and Co-Simulation (V2.0)*. 2014.

[69] T Blochwitz et al. “The Functional Mockup Interface for Tool independent Exchange of Simulation Models”. In: *8th International Modelica Conference 2011* (2009), pp. 173–184. DOI: 10.3384/ecp12076173. URL: http://www.ep.liu.se/ecp{_}article/index.en.aspx?issue=76;article=17.

[70] Christian Kirches. “Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control”. PhD Thesis. 2010.

[71] Moritz Diehl, H Ferreau, and Niels Haverbeke. “Efficient numerical methods for nonlinear MPC and moving horizon estimation”. In: *Nonlinear Model Predictive Control*. Ed. by L Magni. Springer, 2009, pp. 391–417. ISBN: 978-0-85729-500-2. DOI: 10.1007/978-3-642-01094-1{_}32.

[72] I Bauer et al. “ASTRONOMY AND Simulation of chemical reactions and dust destruction in protoplanetary accretion disks”. In: *Astronomy and astrophysics* 289 (1997), pp. 273–289.

[73] Moritz Diehl et al. “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations”. In: *Journal of Process Control* 12 (2002), pp. 577–585. URL: www.elsevier.com/locate/jprocont.

[74] Moritz Diehl, Hans Georg Bock, and Johannes P. Schloeder. “A real-time iteration scheme for nonlinear optimization in optimal feedback control”. In: *SIAM Journal on Control and Optimization* 43.5 (2005), pp. 1714–1736. DOI: 10.1137/S0363012902400713.

[75] Manuel Gräber. “Energieoptimale Regelung von Kälteprozessen”. PhD Thesis. TU Braunschweig, 2014, p. 196.

[76] P O M Scokaert, D Q Mayne, and J B Rawlings. “Suboptimal Model Predictive Control (Feasibility Implies Stability)”. In: *IEEE Transactions on Automatic Control* 44.3 (1999), pp. 648–654.

[77] Moritz Diehl, Rolf Findeisen, and Frank Allg. “A Stabilizing Real-time Implementation of Nonlinear Model Predictive Control”. In: *Real-time and Online PDE-constrained Optimziation*. Ed. by Lorenz T. Biegler et al. SIAM. 2007, pp. 23–52.

[78] Roland Kossel et al. “Effects of Tool Coupling on Transient Simulation of a Mobile Air-Conditioning Cycle Co-simulation”. In: *Proceeedings 7th Modelica Conference*. Como, 2009, pp. 20–22. DOI: 10.3384/ecp09430064.

[79] Eric L Haseltine and James B Rawlings. *A Critical Evaluation of Extended Kalman Filtering and Moving Horizon Estimation*. Tech. rep. Texas-Wisconsin Modeling and Control Consortium, 2003, p. 48.

Appendix A

LT2 Dymola Model

Thermal port:

```
connector FluidPort_nopressure
import SI = Modelica.SIunits;
flow SI.MassFlowRate mDot "MassFlowRate";
SI.ThermodynamicTemperature T "Temperature";
end FluidPort_nopressure;
```

Electrical port:

```
connector ElectricalPort
import SI = Modelica.SIunits;
flow SI.ElectricCurrent I "Current [A]";
SI.Voltage U "Voltage [V]";
end ElectricalPort;
```

smoothTransition **functions from the TIL library:**

```
function smoothTransition
input Real x;
input Real transitionPoint=1;
input Real transitionLength=1;

output Real weigthingFactor;
protected
Real phi "Phase";
parameter Integer funcNum=0 "Function selector";
algorithm
if (x < transitionPoint - 0.5*transitionLength) then
weigthingFactor := 1;
elseif (x < transitionPoint + 0.5*transitionLength) then
phi := (x - transitionPoint)*Modelica.Constants.pi/transitionLength;
if (funcNum == 0) then
weigthingFactor := -1.0/2.0*sin(phi) + 1.0/2.0;
elseif (funcNum == 1) then
weigthingFactor := -1.0/2.0*(2*cos(phi)*sin(phi) + 2*phi - Modelica.Constants.pi)
/Modelica.Constants.pi;
elseif (funcNum == 2) then
weigthingFactor := 1.0/6.0*(-4.0*sin(phi)*cos(phi)^3 - 6.0*phi - 3.0*
sin(2.0*phi) + 3.0*Modelica.Constants.pi)/Modelica.Constants.pi;
elseif (funcNum == 3) then
weigthingFactor := 1.0/30.0*(-16*cos(phi)^5*sin(phi) - 20*cos(phi)^3*
sin(phi) - 30*cos(phi)*sin(phi) - 30*phi + 15*Modelica.Constants.pi)
/Modelica.Constants.pi;
else
weigthingFactor := 0;
end if;
else
weigthingFactor := 0;
end if;
```

```
end smoothTransition;
```

Car_data:

```
model Car_data
Modelica.Blocks.Interfaces.RealInput Torque_effective_Nm;
Modelica.Blocks.Interfaces.RealInput Angular_speed_rpms;
Modelica.Blocks.Interfaces.RealInput Power_demanded_HV_W;
end Car_data;
```

Ambient:

```
model Ambient

import SI = Modelica.SIunits;
parameter SI.Temperature T_ini_BAT_Kreis=50 + 273.15;
parameter SI.Temperature T_ini_LE_Kreis=50 + 273.15;
parameter SI.Pressure p=1e5;
Modelica.Blocks.Interfaces.RealInput T;
end Ambient;
```

drivingCycle:

```
model drivingCycle
Modelica.Blocks.Interfaces.RealInput v_Car_kmh;
end drivingCycle;
```

Control:

```
model Control
Modelica.Blocks.Interfaces.RealInput PWM_BAT;
Modelica.Blocks.Interfaces.RealInput PWM_PE;
Modelica.Blocks.Interfaces.RealInput COOLER_VALVE;
Modelica.Blocks.Interfaces.RealInput CIRCUIT_VALVE;
Modelica.Blocks.Interfaces.RealInput CHILLER_VALVE;
Modelica.Blocks.Interfaces.RealInput PWM_FAN;
end Control;
```

Goal_function:

```
model Goal_function

outer Components.LV_System lV_System;
outer Components.PowerElectronics PowerElectronics;
outer Components.Chiller chiller;
```

```
outer Components.Battery battery;

Real T_battery_degC;
Real T_powerelectronics_degC;
Real Power;

equation

Power = (lV_System.Total_Consumption/PowerElectronics.eta_DCDC) +
chiller.Power_demanded_Chiller_W;
T_battery_degC = battery.T_component - 273.15;
T_powerelectronics_degC = PowerElectronics.T_component - 273.15;

end Goal_function;
```

Pump_PE (Pump_BAT is analog):

```
model Pump_LE_new

// Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet;
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort; // To the LV-net

// Current look-up table declared with 3 inputs
parameter String MAP_Filename_Pump_I="Kennfelder/Pump_I.mat";
Numerical.Spline_Interpolation1D2D3D Pump_I(nin=3, fileName=
MAP_Filename_Pump_I);

// Volume flow rate look-up table for the case "One circuit mode" and
//cooler active declared with 3 inputs
parameter String MAP_Filename_D1_Vdot_le_lmin=
"Kennfelder/D1_Vdot_le_lmin.mat";
Numerical.Spline_Interpolation1D2D3D D1_Vdot_le_lmin(nin=3, fileName=
MAP_Filename_D1_Vdot_le_lmin);

// Volume flow rate look-up table for the case "Two circuit mode" and
//cooler active declared with 3 inputs
parameter String MAP_Filename_D2_Vdot_le_lmin=
"Kennfelder/D2_Vdot_le_lmin.mat";
Numerical.Spline_Interpolation1D2D3D D2_Vdot_le_lmin(nin=3, fileName=
MAP_Filename_D2_Vdot_le_lmin);

// Volume flow rate look-up table for the case "One circuit mode" and
//cooler bypassed declared with 3 inputs
parameter String MAP_Filename_B1_Vdot_le_lmin=
"Kennfelder/B1_Vdot_le_lmin.mat";
Numerical.Spline_Interpolation1D2D3D B1_Vdot_le_lmin(nin=3, fileName=
MAP_Filename_B1_Vdot_le_lmin);

// Volume flow rate look-up table for the case "Two circuit mode" and
//cooler bypassed declared with 3 inputs
parameter String MAP_Filename_B2_Vdot_le_lmin=
```

```
"Kennfelder/B2_Vdot_le_lmin.mat";
Numerical.Spline_Interpolation1D2D3D B2_Vdot_le_lmin(nin=3, fileName=
MAP_Filename_B2_Vdot_le_lmin);

// Other components call
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;
outer Components.Control control;
outer Components.Battery battery;
outer Components.Pump_BAT_new pump_BAT;
outer Components.Charger charger;

// Variables declarations
Real Vdot_lmin;
Real Vdot_D1_lmin;
Real Vdot_D2_lmin;
Real Vdot_B1_lmin;
Real Vdot_B2_lmin;
Real T_average1;
Real T_average2;
Real T_average;

// Calculate thermodynamic properties with TIL Media in the inlet
TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T,
computeTransportProperties=true);

// Calculate thermodynamic properties with TIL Media in the outlet
TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
T=outlet.T,
p=ambient.p,
computeTransportProperties=true);

equation

// Get the current given PWM, Vdot and T
Pump_I.u[1] = control.PWM_PE;
Pump_I.u[2] = Vdot_lmin;
Pump_I.u[3] = T_average - 273.15;
Pump_I.y = LVPort.I;

// Get the Volume flow rate for the case One circuit mode cooler active
D1_Vdot_le_lmin.u[1] = control.PWM_BAT;
D1_Vdot_le_lmin.u[2] = control.PWM_PE;
D1_Vdot_le_lmin.u[3] = T_average - 273.15;
D1_Vdot_le_lmin.y = Vdot_D1_lmin;

// Get the Volume flow rate for the case Two circuit mode cooler active
D2_Vdot_le_lmin.u[1] = control.PWM_BAT;
D2_Vdot_le_lmin.u[2] = control.PWM_PE;
D2_Vdot_le_lmin.u[3] = T_average - 273.15;
```

```
D2_Vdot_le_lmin.y = Vdot_D2_lmin;

// Get the Volume flow rate for the case One circuit mode cooler bypassed
B1_Vdot_le_lmin.u[1] = control.PWM_BAT;
B1_Vdot_le_lmin.u[2] = control.PWM_PE;
B1_Vdot_le_lmin.u[3] = T_average - 273.15;
B1_Vdot_le_lmin.y = Vdot_B1_lmin;

// Get the Volume flow rate for the case Two circuit mode cooler bypassed
B2_Vdot_le_lmin.u[1] = control.PWM_BAT;
B2_Vdot_le_lmin.u[2] = control.PWM_PE;
B2_Vdot_le_lmin.u[3] = T_average - 273.15;
B2_Vdot_le_lmin.y = Vdot_B2_lmin;

// Calculate an average coolant temperature for both circuits, the PE and
// the BAT one.
T_average2 = (inlet.T + outlet.T + charger.inlet.T + charger.outlet.T)/4;
T_average1 = (pump_BAT.inlet.T + pump_BAT.outlet.T + battery.inlet.T +
battery.outlet.T)/4;
T_average = (1 - control.CIRCUIT_VALVE)*T_average2 + (control.CIRCUIT_VALVE)
*(T_average1 + T_average2)/2;

// Find the Volume flow rate selecting the right case
Vdot_lmin = (1 - control.CIRCUIT_VALVE)*(1 - control.COOLER_VALVE)*
Vdot_D2_lmin + (1 - control.CIRCUIT_VALVE)*(control.COOLER_VALVE)*
Vdot_B2_lmin + (control.CIRCUIT_VALVE)*(control.COOLER_VALVE)*
Vdot_B1_lmin + (control.CIRCUIT_VALVE)*(1 - control.COOLER_VALVE)*
Vdot_D1_lmin;

Vdot_lmin*fluidIn.d/60000 = inlet.mDot;

// Mass flow continuity equation
outlet.mDot = -inlet.mDot;
// Coolant is not heated or cooled down in the pump
outlet.T = inlet.T;

end Pump_LE_new;
```

Junction1 (T-Junction just after the cooler):

```
model Valve1

import SI = Modelica.SIunits;
outer Components.Ambient ambient;

\\ Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // from bypass
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // from cooler
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // to Junction2

initial equation

portB.T = ambient.T_ini_LE_Kreis;
```

```
equation

portA.mDot + portB.mDot + portC.mDot = 0;

portA.T*portA.mDot + portB.T*portB.mDot + portC.T*portC.mDot = 0.1*der(
portB.T);

end Valve1;
```

Junction2 (T-Junction on the bottom of Junction1):

```
model Junction2

import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.Control control;

\\ Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // from Junction1
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // to Junction4
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // to Junction3

equation

portA.mDot + portB.mDot + portC.mDot = 0;
portA.T = portB.T;
portC.T = portB.T;

end Junction2;
```

Junction3 (T-Junction on the bottom of Junction2):

```
model Junction3

outer Components.Ambient ambient;
outer Components.Control control;

\\ Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // from Junction2
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // from Close-loop
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // to Pump BAT

initial equation

portC.T = ambient.T_ini_BAT_Kreis;

equation

portC.mDot + portB.mDot + portA.mDot = 0;
portA.T*portA.mDot + portB.T*portB.mDot + portC.T*portC.mDot = 0.1*der(
```

```
portC.T);

end Junction3;
```

Junction4 (T-Junction just in the bottom of the PE pump):

```
model Junction4

import SI = Modelica.SIunits;
outer Components.Ambient ambient;

\\ Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // from Junction2
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // to Pump LE
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // from BAT circuit

initial equation

portB.T = ambient.T_ini_LE_Kreis;

equation

portA.mDot + portB.mDot + portC.mDot = 0;

portA.T*portA.mDot + portB.T*portB.mDot + portC.T*portC.mDot = 0.1*der(
portC.T);

end Junction4;
```

Solenoid valve (cooler valve):

```
model Cooler_Valve

// Connectors
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort; // To LV net
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // From charger
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // To bypass
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // To cooler

// Look-up tables declaration
parameter String MAP_Filename_D1_Vdot_cooler_lmin=
"Kennfelder/D1_Vdot_cooler_lmin.mat";
Numerical.Spline_Interpolation1D2D3D D1_Vdot_cooler_lmin(nin=3, fileName=
MAP_Filename_D1_Vdot_cooler_lmin);

parameter String MAP_Filename_D2_Vdot_cooler_lmin=
"Kennfelder/D2_Vdot_cooler_lmin.mat";
Numerical.Spline_Interpolation1D2D3D D2_Vdot_cooler_lmin(nin=3, fileName=
MAP_Filename_D2_Vdot_cooler_lmin);

// Reference to other components
```

```
import SI = Modelica.SIunits;
outer Components.Control control;
outer Components.Ambient ambient;
outer Components.Battery battery;
outer Components.Pump_BAT_new pump_BAT;
outer Components.Charger charger;
outer Components.Pump_LE_new pump_LE;

// Variables declaration
Real Ploss_Valve_W;
Real Vdot_D1_lmin;
Real Vdot_D2_lmin;
Real Vdot_lmin;
Real T_average1;
Real T_average2;
Real T_average;
Real wf_Valve_Consumption;

// Thermodynamic properties calculations
TILMedia.Liquid_pT outletC(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
computeTransportProperties=true,
p=ambient.p,
T=portC.T);
equation

// Access to the look-up tables
D1_Vdot_cooler_lmin.u[1] = control.PWM_BAT;
D1_Vdot_cooler_lmin.u[2] = control.PWM_PE;
D1_Vdot_cooler_lmin.u[3] = T_average - 273.15;
D1_Vdot_cooler_lmin.y = Vdot_D1_lmin;

D2_Vdot_cooler_lmin.u[1] = control.PWM_BAT;
D2_Vdot_cooler_lmin.u[2] = control.PWM_PE;
D2_Vdot_cooler_lmin.u[3] = T_average - 273.15;
D2_Vdot_cooler_lmin.y = Vdot_D2_lmin;

// Average temperature calculation
T_average2 = (pump_LE.inlet.T + pump_LE.outlet.T + charger.inlet.T +
charger.outlet.T)/4;
T_average1 = (pump_BAT.inlet.T + pump_BAT.outlet.T + battery.inlet.T +
battery.outlet.T)/4;

T_average = (1 - control.CIRCUIT_VALVE)*T_average2 + (control.CIRCUIT_VALVE)
*(T_average1 + T_average2)/2;

// Volume flow rate
Vdot_lmin = (1 - control.CIRCUIT_VALVE)*(1 - control.COOLER_VALVE)*
Vdot_D2_lmin + (1 - control.CIRCUIT_VALVE)*(control.COOLER_VALVE)*0 + (
control.CIRCUIT_VALVE)*(control.COOLER_VALVE)*0 + (control.CIRCUIT_VALVE)
*(1 - control.COOLER_VALVE)*Vdot_D1_lmin;

// Thermal-hydraulic equations
Vdot_lmin*outletC.d/60000 = -portC.mDot;
portA.mDot + portB.mDot + portC.mDot = 0;
```

```
portA.T = portB.T;
portC.T = portA.T;

// Electric equations
wf_Valve_Consumption = 1 - NMPC_lib.Utilities.smoothTransition(
control.COOLER_VALVE,
0.5,
0.0001);

LVPort.I = 15.4*(control.COOLER_VALVE)/LVPort.U;
Ploss_Valve_W = LVPort.U*LVPort.I;

end Cooler_Valve;
```

Solenoid valve (circuit valve):

```
model Circuit_Valve

// Connectors
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort; // To the LV-net
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portA; // From the battery
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portB; // To the BAT circuit
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure portC; // To the PE circuit

// Other components call
import SI = Modelica.SIunits;
outer Components.Control control;

// Variables declaration
Real Ploss_Valve_W;
Real wf_Valve_Consumption;

equation

// Thermal and hidraulic equations

portC.T = portA.T;
portB.T = portC.T;
portB.mDot = -(1 - control.CIRCUIT_VALVE)*portA.mDot;
portC.mDot = -control.CIRCUIT_VALVE*portA.mDot;

// Electric equations

wf_Valve_Consumption = 1 - NMPC_lib.Utilities.smoothTransition(
control.CIRCUIT_VALVE,
0.5,
0.0001);

LVPort.I = 15.4*(control.CIRCUIT_VALVE)/LVPort.U;
Ploss_Valve_W = LVPort.U*LVPort.I;

;
```

```
end Circuit_Valve;
```

Chiller:

```
model Chiller

//connectors
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort; // To the LV-net
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet; // From the BAT pump
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet; // To the BAT

//other components call
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;
outer Components.Control control;
outer Components.Cooler cooler;

//tables
parameter String MAP_Filename_COP_Chiller="Kennfelder/COP_Chiller.mat";
Numerical.Spline_Interpolation1D2D3D COP_Chiller(nin=1, fileName=
MAP_Filename_COP_Chiller);

//declarations
parameter Real cp_component_JkgK=800;
parameter Real m_component_kg=0.44;
parameter Real A_component_m2=0.02;
Real Power_demanded_Chiller_W;
parameter Real PT1_Time_Chiller=40;
Real Qdot_Chiller_W;
Real Ploss_Valve_W;
flow SI.HeatFlowRate Qdot_thermalMass_W;
flow SI.HeatFlowRate Qdot_ambient_W;
flow SI.HeatFlowRate Qdot_fluid_W;
SI.Temperature T_component;
SI.Temperature T_ambient=ambient.T;
Real COP;

//convection and radiation coefficients calculations
Auxiliar.Coefficient_calculations exterior(v_Coolair_kmh=drivingCycle.v_Car_kmh,
T_component=T_component);

//calculate thermodynamic properties
TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T,
computeTransportProperties=true);

TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
T=outlet.T,
p=ambient.p);
```

```
initial equation

T_component = ambient.T_ini_BAT_Kreis;

equation

//Constant Heat transfer of 1350 W when chiller activated
der(Qdot_Chiller_W) = (-1350*control.CHILLER_VALVE - Qdot_Chiller_W)/
PT1_Time_Chiller;

//Chiller electric consumption
COP_Chiller.u[1] = cooler.mdot_air_kgs;
COP_Chiller.y = COP;
Power_demanded_Chiller_W = -Qdot_Chiller_W/COP;

//Chiller valve electric consumption
LVPort.I = 15.4*(1 - control.CHILLER_VALVE)/LVPort.U;
Ploss_Valve_W = LVPort.U*LVPort.I;

//Mass flow balance
outlet.mDot = -inlet.mDot;

//Thermal balance

Qdot_Chiller_W = Qdot_thermalMass_W + Qdot_ambient_W + Qdot_fluid_W;

T_component = fluidOut.T;
Qdot_thermalMass_W = cp_component_JkgK*m_component_kg*der(T_component);
Qdot_ambient_W = A_component_m2*(exterior.alpha*(T_component - T_ambient)
+ exterior.radiation*(T_component^4 - T_ambient^4));
Qdot_fluid_W = inlet.mDot*(fluidOut.h - fluidIn.h);

end Chiller;
```

Radiation and Convection Coefficient calculations:

```
model Coefficient_calculations

//other components call
outer Components.Ambient ambient;

// Inputs speed of the cool air and temperature of the component
input Real v_Coolair_kmh;
input Real T_component;

//Outputs coefficients of heat tranfer
output Real alpha "Convection coefficient";
output Real radiation "Radiation coefficient";

//declarations
parameter Real l_m=1.5;
parameter Real Nu_0=2;
```

```
Real Nu_lam;
Real Nu_turb;
Real Re;
Real Nu;

//call TILMedia to calculate air properties
TILMedia.Gas_pT umgebung(
p=ambient.p,
computeTransportProperties=true,
redeclare TILMedia.GasTypes.TILMedia_MoistAir gasType,
T=(ambient.T + T_component)/2);

//radiation
Real Sigma=5.670e-8 "Boltzmann-Konstante Sigma [W/m2K]";
parameter Real epsilon=0.8;

equation

//convection losses
Re = ((v_Coolair_kmh + 1)*l_m*umgebung.d)/umgebung.transp.eta;
Nu_lam = 0.664*sqrt(Re)*umgebung.transp.Pr^(1/3);
Nu_turb = 0.037*Re^(0.8)*umgebung.transp.Pr/(1 + 2.443*(Re)^(-0.1)*(
umgebung.transp.Pr^(0.66666) - 1));
Nu = Nu_0 + sqrt(Nu_lam^2 + Nu_turb^2);
alpha = (Nu*umgebung.transp.lambda/l_m);

//radiation losses
radiation = epsilon*Sigma;

end Coefficient_calculations;
```

Cooler:

```
model Cooler

//Connectors
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet;
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort;

//Tables access
parameter String MAP_Filename_Heat="Kennfelder/Heat_Coolair.mat";
Numerical.Spline_Interpolation1D2D3D Heat(nin=2, fileName=
MAP_Filename_Heat);

//Other components call
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;
outer Components.Control control;

//Declarations
parameter Real cp_component_JkgK=800;
```

```
parameter Real m_component_kg=1;
parameter Real A_component_m2=0.00001;
flow SI.HeatFlowRate Qdot_thermalMass_W;
flow SI.HeatFlowRate Qdot_fluid_W;
flow SI.HeatFlowRate Qdot_ambient_W;
flow SI.HeatFlowRate Qdot_air_W;
SI.Temperature T_component;
SI.Temperature T_ambient=ambient.T;
Real mdot_air_kgs;
Real v;
Real auxiliar_pwm;
Real Plosses_Fan_W;

//Convection and radiation coefficients calculations
Auxiliar.Coefficient_calculations exterior(v_Coolair_kmh=drivingCycle.v_Car_kmh,
T_component=T_component);

//Calculate thermodynamic properties
TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T,
computeTransportProperties=true);

TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
T=outlet.T,
p=ambient.p,
computeTransportProperties=false);


initial equation

T_component = ambient.T_ini_LE_Kreis;

//Equations
equation

v = drivingCycle.v_Car_kmh;

control.PWM_FAN = auxiliar_pwm*(95.80 - 0.61) + 0.61;


mdot_air_kgs = (1 - auxiliar_pwm)*(9.6190e-10*(v^4) - 4.2631e-07*(v^3) +
6.4034e-05*(v^2) - 8.0918e-04*v - 1.4175e-12) + auxiliar_pwm*(-6.5563e-08
*(v^3) + 2.5577e-05*(v^2) - 1.1911e-04*v + 7.6989e-02);

Heat.u[1] = inlet.mDot*60000/fluidIn.d;
Heat.u[2] = mdot_air_kgs;

outlet.mDot = -inlet.mDot;

-Qdot_fluid_W = Qdot_thermalMass_W + Qdot_air_W + Qdot_ambient_W;
Qdot_air_W = Heat.y*(inlet.T - ambient.T);
```

```
T_component = fluidOut.T;
Qdot_thermalMass_W = cp_component_JkgK*m_component_kg*der(T_component);
Qdot_ambient_W = A_component_m2*(exterior.alpha*(T_component - T_ambient)
+ exterior.radiation*(T_component^4 - T_ambient^4));
Qdot_fluid_W = inlet.mDot*(fluidOut.h - fluidIn.h);

Plosses_Fan_W = LVPort.U*LVPort.I;
LVPort.I = 8.0620e-10*(control.PWM_FAN^6) - 2.0254e-07*(control.PWM_FAN^5)
+ 1.8801e-05*(control.PWM_FAN^4) - 7.5191e-04*(control.PWM_FAN^3) +
1.4589e-02*(control.PWM_FAN^2) - 2.4093e-03*control.PWM_FAN +
1.6431e-03;
end Cooler;
```

HV Battery:

```
model Battery

//Connectors
NMPC_lib.Connectors.BaseClasses.HV_ElectricalPort port;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet;

//Call other components
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;

//Look-up tables
parameter String MAP_Filename_Voltage_Battery=
"Kennfelder/Voltage_Battery.mat";
Numerical.Spline_Interpolation1D2D3D Voltage_Battery(nin=2, fileName=
MAP_Filename_Voltage_Battery);

parameter String MAP_Filename_Ri_Entladen="Kennfelder/Ri_Entladen.mat";
Numerical.Spline_Interpolation1D2D3D Ri_Entladen(nin=2, fileName=
MAP_Filename_Ri_Entladen);

parameter String MAP_Filename_Ri_Laden="Kennfelder/Ri_Laden.mat";
Numerical.Spline_Interpolation1D2D3D Ri_Laden(nin=2, fileName=
MAP_Filename_Ri_Laden);

//Declarations
parameter Real cp_component_JkgK=1500;
parameter Real m_component_kg=120;
parameter Real A_component_m2=0.05;
Real wf_Ri;
Real Ri_Battery_Ohm;
Real I_Battery_A;
Real U_Battery_V;
Real E_Battery_kWh;
Real Pel_Battery_W;
Real SOC_Battery_1;
Real Ploss_Battery_W;
```

```
parameter Real E_Battery_Ini_kWh=6.85;
flow SI.HeatFlowRate Qdot_thermalMass_W;
flow SI.HeatFlowRate Qdot_ambient_W;
flow SI.HeatFlowRate Qdot_fluid_W;
SI.Temperature T_component;
SI.Temperature T_ambient=ambient.T;

//Calculation of convection and radiation components
Auxiliar.Coefficient_calculations exterior(v_Coolair_kmh=drivingCycle.v_Car_kmh,
T_component=T_component);

//Calculate thermodynamic properties
TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T);
TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
T=outlet.T,
p=ambient.p);

initial equation

E_Battery_kWh = E_Battery_Ini_kWh;
T_component = ambient.T_ini_BAT_Kreis;

equation

//Calculation of Voltage depending on SOC and Temperature
Voltage_Battery.u[1] = SOC_Battery_1*100;
Voltage_Battery.u[2] = T_component - 273.15;
Voltage_Battery.y = U_Battery_V;

//Calculation of Ri when charging depending on SOC and Temperature
Ri_Laden.u[1] = SOC_Battery_1*100;
Ri_Laden.u[2] = T_component - 273.15;

//Calculation of Ri when discharging depending on SOC and Temperature
Ri_Entladen.u[1] = SOC_Battery_1*100;
Ri_Entladen.u[2] = T_component - 273.15;

U_Battery_V = port.U;
I_Battery_A = port.I;

Pel_Battery_W = U_Battery_V*I_Battery_A;

//Selection of Ri depending on chargin or discharging

wf_Ri = Numerical.smoothTransition(
Pel_Battery_W,
0,
0.1);

Ri_Battery_Ohm = ((1 - wf_Ri)*Ri_Laden.y + wf_Ri*Ri_Entladen.y)/1000;
```

```
//Power and Energy calculations

Ploss_Battery_W = Ri_Battery_Ohm*I_Battery_A^2;

der(E_Battery_kWh*1000*3600) = Pel_Battery_W - Ploss_Battery_W;

// From measurement the minimum and maximum energy points (E_kWh,SOC in %)
// are known: (0.6, 22) and (6.9,93.2)
SOC_Battery_1 = (-0.597*(E_Battery_kWh^2) + 15.8*E_Battery_kWh + 11.8)/
100;

outlet.mDot = -inlet.mDot;

Ploss_Battery_W = Qdot_thermalMass_W + Qdot_ambient_W + Qdot_fluid_W;

T_component = fluidOut.T;

Qdot_thermalMass_W = cp_component_JkgK*m_component_kg*der(T_component);
Qdot_ambient_W = A_component_m2*(exterior.alpha*(T_component - T_ambient)
+ exterior.radiation*(T_component^4 - T_ambient^4));
Qdot_fluid_W = inlet.mDot*(fluidOut.h - fluidIn.h);

end Battery;
```

Low voltage net:

```
model LV_System

Real Total_Consumption;
parameter Real I_NV_noNT2_Car_A=15;
parameter Real U_network;

NMPC_lib.Connectors.BaseClasses.ElectricalPort LV_demand;
NMPC_lib.Connectors.BaseClasses.ElectricalPort LV_offer;

equation
LV_offer.I = -LV_demand.I + I_NV_noNT2_Car_A;
LV_offer.U = LV_demand.U;
LV_offer.U = U_network;

Total_Consumption = LV_offer.U*LV_offer.I;

end LV_System;
```

Power Electronics:

```
model PowerElectronics

//Connectors
NMPC_lib.Connectors.BaseClasses.HV_ElectricalPort batteryPort;
NMPC_lib.Connectors.BaseClasses.ElectricalPort LVPort;
```

```
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet;


//Other components call
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;
outer Components.Chiller chiller;
outer Components.Car_data car;

//Tables
parameter String MAP_Filename_PlossMot=
"Kennfelder/security_Ploss_PE_Motor.mat";
Numerical.Spline_Interpolation1D2D3D Ploss_PE_Motor(nin=3, fileName=
MAP_Filename_PlossMot)

parameter String MAP_Filename_PlossGen=
"Kennfelder/security_Ploss_PE_Generator.mat";
Numerical.Spline_Interpolation1D2D3D Ploss_PE_Generator(nin=3, fileName=
MAP_Filename_PlossGen)

//Declarations
parameter Real cp_component_JkgK=800;
parameter Real m_component_kg=9.44;
parameter Real A_component_m2=0.04;
parameter Real eta_DCDC=0.9;
Real M_effective_Nm;
Real n_rpms;
Real P_demanded_HV_W;
Real Pel_Battery_W;
Real Pel_LV_W;
Real Ploss_PE_HV_W;
Real Ploss_PE_LV_W;
Real Ploss_PE_W;
Real wf_Ploss;
flow SI.HeatFlowRate Qdot_thermalMass_W;
flow SI.HeatFlowRate Qdot_ambient_W;
flow SI.HeatFlowRate Qdot_fluid_W;
SI.Temperature T_component;
SI.Temperature T_ambient=ambient.T;

//Convection & radiation calculations
Auxiliar.Coefficient_calculations exterior(v_Coolair_kmh=drivingCycle.v_Car_kmh,
T_component=T_component);


TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T,
computeTransportProperties=true);

TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
```

```
T=outlet.T,
p=ambient.p);

initial equation

T_component = ambient.T_ini_LE_Kreis;

equation


M_effective_Nm = car.Torque_effective_Nm;
n_rpms = car.Angular_speed_rpms;
P_demanded_HV_W = car.Power_demanded_HV_W + chiller.Power_demanded_Chiller_W;


Pel_LV_W = LVPort.U*LVPort.I;
Pel_Battery_W = batteryPort.U*batteryPort.I;

Ploss_PE_Motor.u[1] = M_effective_Nm;
Ploss_PE_Motor.u[2] = n_rpms;
Ploss_PE_Motor.u[3] = batteryPort.U;

Ploss_PE_Generator.u[1] = M_effective_Nm;
Ploss_PE_Generator.u[2] = n_rpms;
Ploss_PE_Generator.u[3] = batteryPort.U;


wf_Ploss = Numerical.smoothTransition(
M_effective_Nm,
0,
0.1);

Ploss_PE_HV_W = -(wf_Ploss)*Ploss_PE_Generator.y - (1 - wf_Ploss)*
Ploss_PE_Motor.y;

Ploss_PE_LV_W = Pel_LV_W*(1 - eta_DCDC)/eta_DCDC;

Ploss_PE_W = (-Ploss_PE_HV_W - Ploss_PE_LV_W);

-Pel_Battery_W = +Pel_LV_W - Ploss_PE_W - P_demanded_HV_W;

outlet.mDot = -inlet.mDot;


Ploss_PE_W = Qdot_thermalMass_W + Qdot_ambient_W + Qdot_fluid_W;

T_component = fluidOut.T;
Qdot_thermalMass_W = cp_component_JkgK*m_component_kg*der(T_component);
Qdot_ambient_W = A_component_m2*(exterior.alpha*(T_component - T_ambient)
+ exterior.radiation*(T_component^4 - T_ambient^4));
Qdot_fluid_W = inlet.mDot*(fluidOut.h - fluidIn.h);
end PowerElectronics;
```

Charger:

```
model Charger

//Other components call
import SI = Modelica.SIunits;
outer Components.Ambient ambient;
outer Components.drivingCycle drivingCycle;
outer Components.Control control;

//Declarations
parameter Real cp_component_JkgK=800;
parameter Real m_component_kg=5.8;
parameter Real A_component_m2=0.04;

flow SI.HeatFlowRate Qdot_thermalMass_W;
flow SI.HeatFlowRate Qdot_ambient_W;
flow SI.HeatFlowRate Qdot_fluid_W;
SI.Temperature T_component;
SI.Temperature T_ambient=ambient.T;

//Convection & Radiation calculations
Auxiliar.Coefficient_calculations exterior(v_Coolair_kmh=drivingCycle.v_Car_kmh,
T_component=T_component);


TILMedia.Liquid_pT fluidIn(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
p=ambient.p,
T=inlet.T,
computeTransportProperties=true);

TILMedia.Liquid_pT fluidOut(
redeclare TILMedia.LiquidTypes.TILMedia_Glysantin_50 liquidType,
T=outlet.T,
p=ambient.p);

NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure inlet;
NMPC_lib.Connectors.BaseClasses.FluidPort_nopressure outlet;

initial equation

T_component = ambient.T_ini_LE_Kreis;

equation


outlet.mDot = -inlet.mDot;

0 = Qdot_thermalMass_W + Qdot_ambient_W + Qdot_fluid_W;

T_component = fluidOut.T;
Qdot_thermalMass_W = cp_component_JkgK*m_component_kg*der(T_component);
Qdot_ambient_W = A_component_m2*(exterior.alpha*(T_component - T_ambient)
+ exterior.radiation*(T_component^4 - T_ambient^4));
```

```
Qdot_fluid_W = inlet.mDot*(fluidOut.h - fluidIn.h);

end Charger;
```

Appendix B

MUSCOD-II files to formulate an OCP

.DAT File example of an optimization in the simulation environment:

```
                                                    

**********************
* # Shooting points   *
*    & Horizon        *
**********************

nshoot
0: 2

* model stage duration start values
*, scale factors, and bounds

h
0: 200

h_sca
0: 200

h_min
0: 200

h_max
0: 200

* specification mode for differential
* state variable start values

s_spec
2

**********************
* Differential states *
*         x           *
**********************

xd_name
0: valve2.portC.T
1: valve4.portC.T
2: valve1.portB.T
3: chiller.Qdot_Chiller_W
4: chiller.T_component
5: charger.T_component
6: PowerElectronics.T_component
7: battery.E_Battery_kWh
8: battery.T_component
9: cooler.T_component
10: SPEED_KMH
11: TORQUE_NM
12: RPMS
13: ELECTRICAL_POWER_W
14: T_OUTSIDE

* start values diff. states

sd(0,S)
0: 2.953500e+02
1: 2.951500e+02
2: 2.953500e+02
3: 0
4: 2.951500e+02
5: 2.953500e+02
6: 2.953500e+02
7: 7
8: 2.951500e+02
9: 2.953500e+02
10: 0
11: 0
12: 0
13: 0
14: 2.938500e+02

* scale values diff. states

sd_sca(*,*)
0: 3.018971e+02
1: 2.999388e+02
2: 3.022089e+02
3: 400
4: 2.996927e+02
5: 3.034365e+02
6: 3.037514e+02
7: 3.383772e+00
8: 2.996457e+02
9: 3.012206e+02
10: 4.316398e+01
11: 1.690294e+02
12: 1.534287e+03
13: 2.557038e+04
14: 2.928295e+02

* min values diff. states

sd_min(*,*)
0: 263.15
1: 263.15
2: 263.15
3: -5000
4: 263.15
5: 263.15
6: 263.15
7: -10
8: 263.15
9: 263.15
10: -10
```

```
11: -500
12: -10000
13: -120000
14: 263.15

* max values diff. states

sd_max(*,*)
0: 333.15
1: 333.15
2: 333.15
3: 5000
4: 333.15
5: 333.15
6: 333.15
7: 8.8
8: 333.15
9: 333.15
10: 200
11: 500
12: 10000
13: 120000
14: 333.15

sd_fix(0,S)
0: 1
1: 1
2: 1
3: 1
4: 1
5: 1
6: 1
7: 1
8: 1
9: 1
10: 1
11: 1
12: 1
13: 1
14: 1

**********************
*   Control signals   *
*         u           *
**********************

* control parameterization

u_type(*)
0: 0
1: 0
2: 0
3: 0
4: 0
5: 0


* control signals

u_name
0: CHILLER_VALVE
1: CIRCUIT_VALVE
2: PWM_FAN
3: COOLER_VALVE
4: PWM_BAT
5: PWM_PE

* start values controls

u(0,*)
0: 0
1: 1
2: 0.61
3: 0
4: 40
5: 40

* control values scaling

u_sca(*,*)
0: 1
1: 1
2: 50
3: 1
4: 40
5: 40

* min control values

u_min(0,*)
0: 0
1: 0
2: 10
3: 0
4: 16
5: 30

* max control values

u_max(0,*)
0: 1
1: 1
2: 90
3: 1
4: 100
5: 100


**********************
*  Tuning parameters  *
**********************
```

```
216
217 p_name
218 0: ambient.T_ini_BAT_Kreis
219 1: ambient.T_ini_LE_Kreis
220 2: battery.E_Battery_Ini_kWh
221 3: coef_T
222 4: coef_P
223 5: time_mean
224
225 p_fix
226 0: 1
227 1: 1
228 2: 1
229 3: 1
230 4: 1
231 5: 1
232
233
234 p
235 0: 2.951500e+02
236 1: 2.953500e+02
237 2: 7
238 3: 100
239 4: 2.500000e+00
240 5: 3160
241
242 p_sca
243 0: 2.951500e+02
244 1: 2.953500e+02
245 2: 7
246 3: 100
247 4: 2.500000e+00
248 5: 3160
249
250 p_min
251 0: 263.15
252 1: 263.15
253 2: 0
254 3: 0
255 4: 0
256 5: 0
257
258
259 p_max
260 0: 333.15
261 1: 333.15
262 2: 9
263 3: 100
264 4: 100
265 5: 1000000
266
267 ***********************
268 * Objective function  *
269 ***********************
270
271 of_sca
272 0: 8
273
274 of_min
275 0
276
277 of_max
278 0: 1.108240e+00
279
280 nhist
281 30
282
283 **********************
284 * TISC settings *
285 **********************
286
287 tisc_ip
288 127.0.0.1
289
290 tisc_syncrate
291 2500000
292
293 tisc_syncMethod
294 2
295
296 tisc_nInitialSyncs
297 0
298
299 tisc_nConvergenceSyncs
300 0
301
302 tisc_nIterations
303 1
304
305 **********************
306 * Choosing libraries *
307 **********************
308
309 libmodel
310 SRC/libNMPC_LT2
311
312 libhessian
313 hess_limitedmemoryupdate
314
315 libsolve
316 solve_fullstep
317
318 libcond
319 cond_std
320
321 libtchk
322 tchk
323
324 libmssqp
325 mssqp_tisc
```

```
326
327 libeval
328 eval_ind
329
330 libind
331 0: ind_daesol
332
333 libqps
334 qps_qpopt
335
336 libplot
337 plot_noplot
338
339 *********************************
340 * Setting algorithmic parameters *
341 *********************************
342
343 options_acc
344 1e-6
345 options_ftol
346 -1.0
347 options_itol
348 -1.0
349 options_rfac
350 0.0
351 options_levmar
352 0.0
353 options_qp_featol
354 1.0e-8
355 options_qp_relax
356 1.1
357 options_nhtopy
358 0
359 options_frstart
360 0
361 options_frmax
362 0
363 options_itmax
364 1000000
365 options_plevel_screen
366 0
367 options_plevel_file
368 0
369 options_plevel_matlab
370 0
371 options_bflag
372 -1
373 options_qp_itmax
374 10000000
375 options_qp_expand
376 99999999
377 options_sflag
378 0
379 options_options_wflag
380 0
381 options_cflag
382 0
383 options_output_ps
384 1
385 options_output_gif
386 0
387 options_iest_hess_plitt
388 0
```

.cpp File example of an optimization in the simulation environment:

```
1
2  #include <cstdio>
3  #include <windows.h>
4  #include "def_usrmod.hpp"
5  #include "fmiModelTypes.h"
6
7  using namespace std;
8  // Define number of tuning parameters & constraints
9  #define  NMOS   1
10 #define  NP     6
11 #define  NRC    0
12 #define  NRCE   0
13
14 // Define number of differential & algebraic states and controls
15 #define  NXD    15
16 #define  NXA    0
17 #define  NU     6
18 #define  NPR    0
19
20 typedef enum  {fmiOK, fmiWarning, fmiDiscard, fmiError, fmiFatal} fmiStatus;
21 typedef void  (*fmiCallbackLogger)(fmiComponent c, fmiString instanceName,
       fmiStatus status,\\fmiString category, fmiString message, ...);
22 typedef void* (*fmiCallbackAllocateMemory)(size_t nobj, size_t size);
23 typedef void  (*fmiCallbackFreeMemory)(void* obj);
24 typedef struct {
25 fmiCallbackLogger         logger;
26 fmiCallbackAllocateMemory allocateMemory;
27 fmiCallbackFreeMemory     freeMemory;
28 } fmiCallbackFunctions;
29 typedef struct {
30 fmiBoolean iterationConverged;
31 fmiBoolean stateValueReferencesChanged;
32 fmiBoolean stateValuesChanged;
33 fmiBoolean terminateSimulation;
34 fmiBoolean upcomingTimeEvent;
35 fmiReal    nextEventTime;
36 } fmiEventInfo;
37
38 HINSTANCE hInstLibrary;
39
40 typedef fmiComponent (*_fmiInstantiateModel)(fmiString instanceName, fmiString
       GUID, fmiCallbackFunctions functions, fmiBoolean loggingOn);
41
42 typedef fmiStatus (*_fmiSetTime)(fmiComponent c, fmiReal time);
43 typedef fmiStatus (*_fmiSetReal)(fmiComponent c, const fmiValueReference vr[],
       size_t nvr, const fmiReal    value[]);
44 typedef fmiStatus (*_fmiInitialize)(fmiComponent c, fmiBoolean
       toleranceControlled, fmiReal relativeTolerance, fmiEventInfo* eventInfo);
45 typedef fmiStatus (*_fmiSetContinuousStates)(fmiComponent c, fmiReal states[],
       size_t nx);
46 typedef fmiStatus (*_fmiGetDerivatives)(fmiComponent c, fmiReal derivatives[],
       size_t nx);
```

```
47 typedef fmiStatus (*_fmiGetReal)(fmiComponent c, const fmiValueReference vr[],
       size_t nvr, fmiReal    value[]);
48
49 typedef void (*_fmiFreeModelInstance)(fmiComponent c);
50
51 _fmiSetTime             fmiSetTime;
52 _fmiSetReal             fmiSetReal;
53 _fmiSetContinuousStates fmiSetContinuousStates;
54 _fmiGetDerivatives      fmiGetDerivatives;
55 _fmiGetReal              fmiGetReal;
56 _fmiFreeModelInstance   fmiFreeModelInstance;
57
58 static void fmuLogger(fmiComponent c, fmiString instanceName, fmiStatus status,
59 fmiString category, fmiString message, ...) {    }
60
61 fmiComponent fmu;
62 const fmiValueReference uRef[NU] = {352321541, 352321539, 352321540, 352321538,
       352321536, 352321537, };
63 const fmiValueReference pRef[NP] = {16777216,16777217,16777218,};
64
65 class InstantiateFMU
66 {
67 public:
68 InstantiateFMU();
69 ~InstantiateFMU();
70 };
71
72 InstantiateFMU::InstantiateFMU()
73 {
74 fmiCallbackFunctions callbacks;
75 fmiStatus status;
76
77 callbacks.logger = fmuLogger;
78 callbacks.allocateMemory = calloc;
79 callbacks.freeMemory = free;
80
81 const char* instanceName = "fmu";
82 const char* GUID = "{c2b39c0c-3e9a-4f19-93d0-e7ff9ba89c4e}";
83
84 // load DLL
85 HINSTANCE hInstLibrary = LoadLibrary("NMPC_LT2.dll");
86 _fmiInstantiateModel fmiInstantiateModel;
87 _fmiInitialize       fmiInitialize;
88 fmiInstantiateModel =    (_fmiInstantiateModel)    GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiInstantiateModel");
89 fmiInitialize =          (_fmiInitialize)          GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiInitialize");
90 fmiSetTime =             (_fmiSetTime)             GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiSetTime");
91 fmiSetReal =             (_fmiSetReal)             GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiSetReal");
92 fmiSetContinuousStates = (_fmiSetContinuousStates) GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiSetContinuousStates");
93 fmiGetDerivatives =      (_fmiGetDerivatives)      GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiGetDerivatives");
```

```
94 fmiFreeModelInstance = (_fmiFreeModelInstance) GetProcAddress(hInstLibrary, "
       NMPC_LT2_fmiFreeModelInstance");
95 fmiGetReal = (_fmiGetReal) GetProcAddress(hInstLibrary, "NMPC_LT2_fmiGetReal"
       );
96
97 // Instantiate fmu
98 fmu = fmiInstantiateModel(instanceName, GUID, callbacks, fmiFalse);
99 if (!fmu) printf("\n fmu instantiation failed! \n");
100
101 // Set Time
102 status = fmiSetTime(fmu, 0.0);
103
104 // Set Inputs
105 const fmiReal uInit[NU] = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
106 status = fmiSetReal (fmu, uRef, NU, uInit);
107
108 // Initialize
109 fmiEventInfo eventInfo;
110 status = fmiInitialize(fmu, fmiFalse, 0.0, &eventInfo);
111 if (status==0) printf("fmu initialized.");
112 }
113
114 InstantiateFMU::~InstantiateFMU()
115 {
116 fmiFreeModelInstance(fmu);
117 }
118
119 InstantiateFMU instantiateFMU;
120
121 // Define type of objective function
122 static void lfcn(double *t, double *xd, double *xa, double *u,
123 double *p, double *lval, double *rwh, long *iwh, long *info)
124 {
125 // Set Time
126 fmiSetTime(fmu, *t);
127
128 // Set Inputs
129 fmiSetReal (fmu, uRef, NU, u);
130
131 // Set Parameters
132 fmiSetReal (fmu, pRef, NP, p);
133
134 // Set States
135 fmiSetContinuousStates(fmu, xd, NXD);
136
137 //Get a variable from the Model that is not a differential state
138
139 double Power[1];
140 const fmiValueReference ref[1] = {905977280}; // value reference from fmu xml
       file
141 fmiGetReal(fmu, ref, 1, Power);
142
143 // OBJECTIVE FUNCTION
144
```

```
145 *lval = p[4]*(Power[0]-200)/(1000*p[5]) + p[3]*((0.000000603*((xd[8]-273.15)*(xd
        [8]-273.15)*(xd[8]-273.15)*(xd[8]-273.15)) - 0.0000267*((xd[8]-273.15)*(xd
        [8]-273.15)*(xd[8]-273.15)) + 0.000348*((xd[8]-273.15)*(xd[8]-273.15)) -
        0.00979*(xd[8]-273.15) + 0.218))/p[5];
146 }
147
148 static void ffcn(double *t, double *xd, double *xa, double *u,
149 double *p, double *rhs, double *rwh, long *iwh, long *info)
150 {
151 // Set Time
152 fmiSetTime(fmu, *t);
153
154 // Set Inputs
155 fmiSetReal (fmu, uRef, NU, u);
156
157 // Set Parameters
158 fmiSetReal (fmu, pRef, NP, p);
159
160 // Set States
161 fmiSetContinuousStates(fmu, xd, NXD);
162
163 // Get Derivatives
164 fmiGetDerivatives(fmu, rhs, NXD);
165 }
166
167 extern "C" void def_model(void);
168 void def_model(void)
169 {
170 def_mdims(NMOS, NP, NRC, NRCE);
171 def_mstage(
172 0,
173 NXD, NXA, NU,
174 NULL, lfcn,
175 0, 0, 0, NULL, ffcn, NULL,
176 NULL, NULL
177 );
178 }
```